INFORMATION CENTRIC UPDATING SCHEME USING EASRC FOR UPPER-LIMB MYOELECTRIC CONTROL

by

Teja Karri

A thesis submitted to The Johns Hopkins University

in conformity with the requirements for the degree of

Master of Science in Engineering

Baltimore, Maryland

May, 2019

Abstract

In this thesis, the idea of an information centric updating scheme for improved upper-limb prosthesis control is explored. The basis for this updating framework is the classifier EASRC - a hybrid classifier that takes advantage of an Extreme Learning Machine (ELM) and Sparse Representation Classification (SRC). Due to its hybrid nature, EASRC is able to perform classification depending on some confidence threshold. If the system is confident, then EASRC uses boundaries defined for each class in order to provide a prediction. Otherwise, it uses the output of ELM as a filter to obtain some smaller subset of potential classes which allows SRC to generate a prediction by input reconstruction. The dependency on input reconstruction places an emphasis on the vector-subspace occupied by each class. The class that contributes the most to the reconstruction is the class that is predicted by SRC. The speed of the ELM and the accuracy of SRC allow EASRC to be robust classifier in the EMG problem space. EASRC, while robust, is still prone to performance degradation due to forces that modify the EMG signal characteristics (nonstationarities). In order to get around this issue, the classifier can be augmented with an updating system to prevent this signal degradation over time. In order to optimize the feature vectors that are included in the classifier, a simple updating scheme that performs K-means on some buffered input data is performed to sample the most representative inputs. These representative inputs are incorporated into the class sub-dictionary after compressing the class sub-dictionary by K members. This method of replacement allows the system to adapt to changing signals caused by nonstationarities. In this thesis, we explore the updating system's classification performance under the limb-position effect when requested to perform certain hand grips in static locations away from the original training site. From the online experiment, we observe statistically significant results suggesting improvement relative to the control classifier with $p < 0.0001$. We suggest this updating system for EASRC may show potential for amputees challenged by their temporally changing EMG data.

Primary Reviewer: Nitish V. Thakor, PhD

Secondary Reviewer: Gene Y. Fridman , PhD

Secondary Reviewer: J. Tilak Ratnanather, PhD

Acknowledgments

Firstly, I would like to thank Professor Nitish Thakor for his time, support, and guidance over the last year. A special thank you to Luke Osborn and Joseph Betthauser for being the big brothers of the lab and for their guidance and insights in my research and for their emotional support. I would also like to thank members Christopher Hunt and Mark Iskarous for their help with troubleshooting all of the hardware and software issues that I have run into and for spending the time to teach me completely alien concepts. Without all of them, I would have certainly been lost and would have struggled through the year a lot more. I would also like to thank partners, Infinite Biomedical Technologies. In particular, CEO Rahul Kaliki and Research Manager George Levay, for their guidance and insights in the problem space. I would also like to thank Professor Gene Fridman and Professor Tilak Ratnanather for spending the time to review through this large document. Last but not least, I would like to thank all of the students and faculty I have interacted with over the year for their support, guidance, and friendship.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

The human hand is one of the most dexterous evolutionary adaptations seen in the animal kingdom. It has evolved to interface with various objects in the world using large coarse movements or fine control - all seamlessly operated by a vast network of neurons that relay intention via signals to the muscles. The utility provided by the human hand and the components that work in unison cannot be overstated, for the lack of even a small appendage such as the thumb requires an amputee to adapt to a new normal.

Unfortunately for upper-limb amputees, no solution currently exists that is capable of capturing the efficacy and reliability afforded by the biological hand. However, yearly improvements and advancements in neuro-engineering has pushed prosthesis technology closer towards recreating a antorhpomorphic mechatronic hand capable of actuating individual fingers in order to perform grasps reflective of activities performed in daily living. In order to consistently control hand actions and retain this control over long periods of time, requires a consistent stream of predictions. This thesis focuses on supplementing a machine learning algorithm with an updating system to make the system robust to changing muscular signals.

1.2 Capturing EMG Data

The current state of the art in prosthesis control is facilitated by using electrical signals acquired from activated muscle. These electromyography (EMG) signals recorded from the skeletal muscle (Fougner et al., 2012) are the primary means of capturing intended motion. As motions generally involve the use of multiple muscles across the body, the location of EMG recordings at the physiologically appropriate muscle can provide greater information of the movement (Young, Hargrove, and Kuiken, 2011). With able-bodied subjects, it is trivial to decode electrical signals as there are a host of suitable electrode sites that can be used for accurate prosthesis control and each site can provide useful context regarding the motion. With amputees however, the difficulty lies in attempting to extract neural information that is unavailable due to damaged musculature. Even still, it is possible to obtain useful EMG information from users with transradial amputations as there is still existing musculature that can be leveraged (Scheme and Englehart, 2011). The more proximal to the shoulder an amputation is, the more muscle, and therefore information, is hence lost. This loss of information content makes it increasingly difficult to provide functionality to a prosthesis. In order to overcome these limitations, a surgical technique developed for transhumeral amputees known as targeted muscle reinnervation (TMR) was used to remap undamaged nerves to the residual limbs of the chest where they can act as signal amplifiers for physiological EMG signals. This way, the information content can be increased by incorporating nerves used for hand/wrist/finger control, even in the absence of muscles themselves, for further prosthesis control (Kuiken et al., 2009). To effectively extract EMG data and supplement surgical methods like TMR, consideration must be given to the number and position of electrodes across the residual limb to target these nerves. Previous studies have shown that prosthesis performance does increase with the addition of more electrodes but with diminishing returns (Hargrove, Englehart, and Hudgins, 2007). Additionally, since amputees have unique physiology that varies from one subject to the next, the number and position of these electrodes will vary (Farina et al., 2010). While several methods exist to record EMG data, surface EMG (sEMG) is conventionally used for signal acquisition because it is non-invasive and provides sufficiently rich neural information for signal processing (Farina et al., 2014). In order to obtain the signals, sEMG electrodes are generally oriented radially at equal distances around the arm while electrode placement is generally chosen to suit the user's specific amputation. The sEMG signals acquired from the respective electrode sites are continuously streamed, while a sliding time window is used to capture a small time-window of the most recent sEMG data. This time-window sEMG data is then subsequently filtered to reduce noise, processed to increase information density, and squeezed into a single feature vector where it is sent to a pattern recognition system in order to generate a prediction of the subjects intended grip. The ability to provide reliable predictions places a heavy focus on pattern recognition which references the streaming feature vectors to its trained model.

1.3 Pattern Recognition

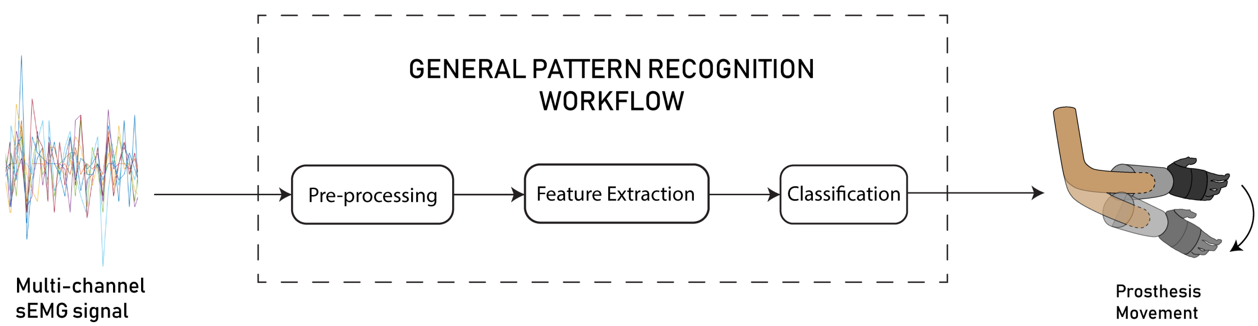

Figure 1.1: The general flow of pattern recognition in the EMG problem space. A multi-channel sEMG signal is first obtained using electrodes placed on the skin of a subject, where the raw signal subsequently gets filtered, feature extracted, and classified. The final prediction is then sent to the prosthesis in order to recreate the intended grasp.

Pattern recognition is a category of computer algorithms tasked with finding similarities in data that can be ultimately used for prediction or recognition of an event. In the EMG problem space, the main goal of pattern recognition is to extract meaningful information from acquired raw EMG data and classify the motion. Broadly speaking, there are two main types of pattern recognition systems employed in EMG prosthesis control: supervised learning and unsupervised learning. The general flow of information as illustrated in Figure 1.1. The difference between supervised and unsupervised learning is that in supervised learning, each of the data points is given an associated label corresponding to a class, or hand grip. This is to aid the algorithm in finding the optimal boundaries to distinguish between the required classes - allowing the algorithm to generate the appropriate predictions. In unsupervised learning, the datum is not given a label, thus, requiring the computer to "learn" the number of classes that exist within the problem space, and then subsequently generate the boundaries that best differentiate the classes in the data space.

1.3.1 Supervised Learning

One of the most commonly used supervised learning algorithms in the EMG problem space is Linear Discriminant Analysis (LDA). LDA has been cited for its "simplicity of implementation and ease of training" (Scheme and Englehart, 2011). The main idea is to determine lower-dimensional hyperplanes that are capable of maximizing class distances in order to generate linear boundaries. The algorithm makes an underlying assumption that the data for each class is normally distributed and that they all have the same covariance.

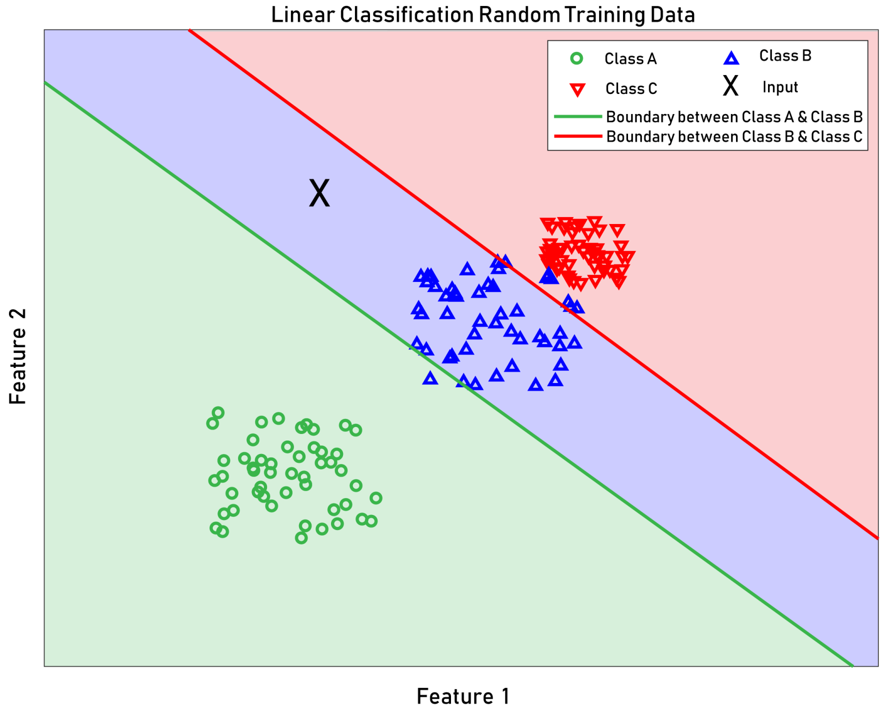

Figure 1.2: A depiction of LDA's boundaries for classes A, B, and C. The highlighted regions correspond to the feature space occupied by the respective colored class divided by the hyperplanes generated after optimization. An input X will be classified depending on the boundaries that it falls under. In this case, input X falls in the blue region and would hence be labeled as class B.

Assume a problem space with m samples and K classes. Where each input $x_t \in \mathbb{R}^n$ and the output $y_t = \{1, ..., K\}$ is in one of the K classes. From the definition of maximum likelihood, we wish to maximize the probability of a particular class given an input: $P(Y = k|X = x)$. In order to obtain this estimate we can rewrite the equation using Bayes Rule as:

$$P(Y = k|X = x) = \frac{P(X = x|Y = k)P(Y = k)}{\sum_{i=1}^{K} P(X = x|Y = i)P(Y = i)}$$

We let $P(X|Y)$ be represented as a Gaussian $\mathcal{N}(\mu, \sum)$ with some mean μ and covariance $\sum$:

$$P(X|Y) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma(x-\mu)\right)$$

We let the priors $P(Y = i) = \pi_i$ under the constraint $\sum_{i=1}^{K} \pi_i = 1$. We then calculate the appropriate mean and covariances to finally optimize for the final max log likelihood of $P(X|Y)P(Y)$

Classes are predicted based on which of the K classes maximizes the log-likelihood given an input x:

$$\arg\max_k \left[-\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\hat{\Sigma}| - \frac{1}{2}(x-\hat{\mu}_k)^T \hat{\Sigma}^{-1}(x-\hat{\mu}_k) + \log \pi_k \right]$$

Geometrically, the classifier can utilize the boundaries demarcated by the hyperplanes in order to label an input with the appropriate class. As a result, any input that lies within a boundary also corresponds to the class that maximizes the log-likelihood of the input.

1.3.2 Unsupervised Learning

Since no true labels are given to the data in unsupervised learning, algorithms are commonly augmented with clustering methods in order to derive class groupings and thus assist in generating class boundaries. One commonly used method known for its simplicity in finding structures within data, is the K-Means clustering algorithm. The idea behind K-Means is to iteratively derive a solution where K-centroids can properly represent the means of K different groups observed in the data space. This iterative process begins by generating K centroids randomly within the space of n observable data points. Each of the n data points, is associated to one of K centroids that has the minimized Euclidean distance. The inputs assigned to each cluster are used to calculate a new centroid location. These steps are cycled until there is no change in the cluster assignment or when the change in cluster position is smaller than some threshold ϵ. This iterative process is visualized in Figure 1.3.

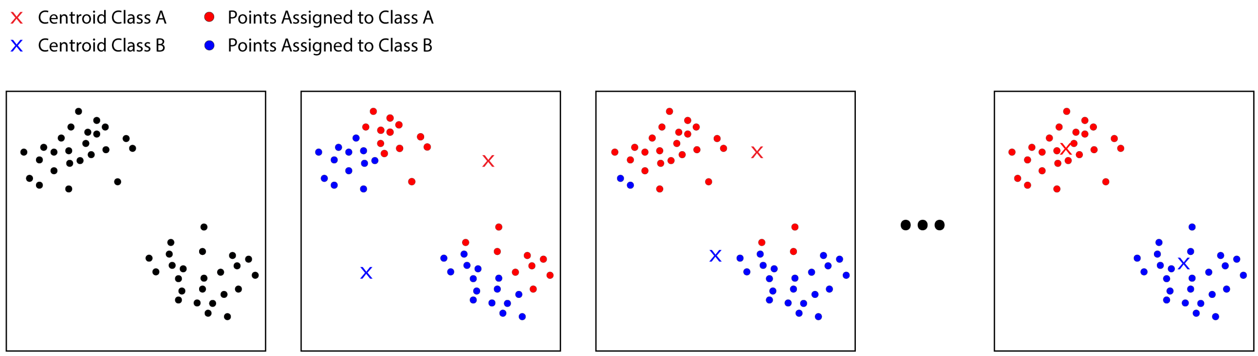

Figure 1.3: K-Means is an clustering algorithm that assigns K-centroids to the given data space. This illustration simply represents the way a K-Means algorithm may approach to the final centroid positions provided randomly initialized centroids, given in X, given the data distribution given by circles. The system converges to the optimum centroid locations over a series of iterations after being randomly initialized. The system stops when the change in centroids is very minimal or nonexistent.

1.4 Signal Processing

Raw data streamed through the electrodes do not have enough information to effectively differentiate signals and be useful for classification. All of the incoming raw data is first preprocessed by zero centering, filtering to eliminate noise, and amplify the signal. After preprocessing the raw signal, the filtered signal is assessed using a sliding window of certain length and step size in order to increase the information density captured within the input. Extracted signal features are then condensed into a single feature vector that contains information regarding the characteristics of the filtered signal. These feature vectors, are what constitute the data space discussed previously, and are used by the classifier in order to optimize class boundaries. An incoming feature vector is referenced with the boundaries of the classifier and is then given a prediction. Raw EMG data and sliding window can be visualized in Figure 1.4.

1.4.1 Preprocessing

Preprocessing EMG data is important to reduce the noise from the input signal and capture physiologically relevant information in order to make reliable predictions. In order

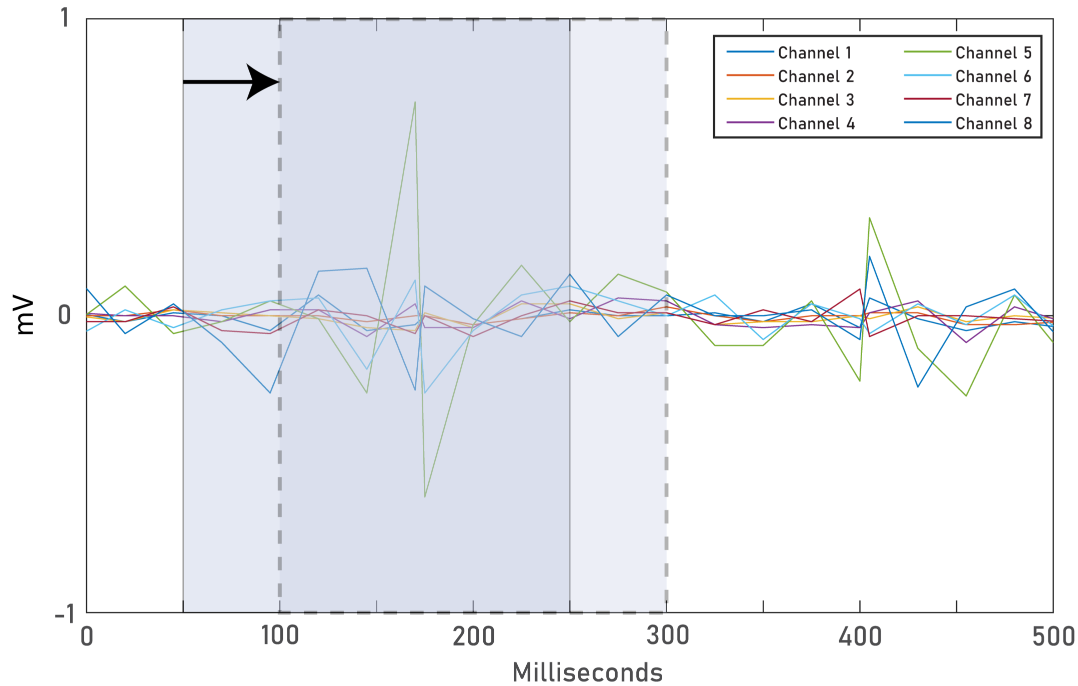

Figure 1.4: This represents pre-processed sEMG data acquired from eight electrodes radially configured about a person's arm. The highlighted region represents 200ms worth of data that is used as a signal input to the classifier at any one point of time. This window is then feature extracted and subsequently classified. The dashed window represents the 50ms window shift that is used to generate a new feature vector containing new data. Notice the signal overlap that exists when shifting the data. At any point during the shift, only 50ms of new data is obtained during a new classification.

to maximize user performance while balancing controller delay, different methods of signal preprocessing have been explored. While greater information is inherently a useful metric to make appropriate classification decisions, too much information processing can result increased controller delay and error rate (Smith et al., 2011). Two of the most important aspects are the filter as well as the size and step of the sliding window.

The filter is optimized to capture most of the power-spectra of the EMG signal. A band-pass filter is used to capture frequencies between 20Hz - 500Hz. While there are minor amounts of energy in sEMG signals below 20Hz, they are unstable and do not provide valuable information in prediction. Power spectra of higher frequencies suggests that there is useful information at that can assist in reducing noise and decreasing movement artifacts (De Luca et al., 2010). An optimum is therefore reached by providing a bandpass filter between 20Hz and 500 Hz to capture a majority of the relevant EMG data. Refer to Figure 1.5 as an example for visualizing the EMG power spectra.

Previous work on finding an optimum window length for EMG signal analysis has suggested a window length between 150ms to 250ms stepping at 25ms increments (Smith et al., 2011). Smith et al. (2011) illustrated by increasing the window length past this optimum, there was a subsequent increase in the controller delay and a decrease in classification error. The increase in temporal information allows for greater context in deciding a class, but it consequently increases the computational time. Notably, Smith et al. (2011) also showed that increasing the number of channels for signal acquisition results in a greater spatial sampling of muscles and consequently more information density, although certain window sizes do not show significant improvement even with greater information density.

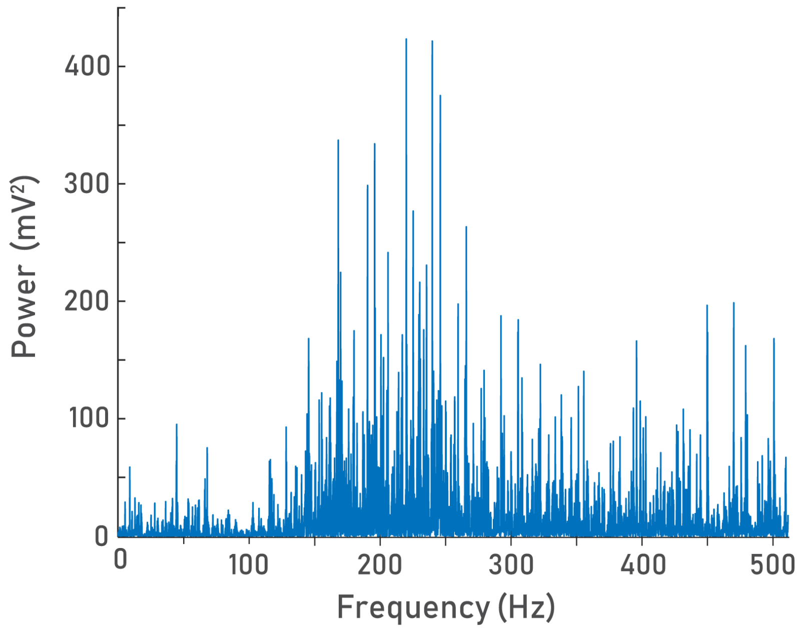

Figure 1.5: This is a visualization of the power spectra from a raw sEMG signal streamed at 1024 Hz. Notice that most of the activity of the signal is captured in the 100 Hz range, while some measurable power is still observed at around 500 Hz. Most of the significant inputs seem to start by about 50 Hz as indicated by this particular power spectra. While this is representative of a single electrode, there can be some variance in the power spectra profiles observed across all electrodes.

1.4.2 Feature Extraction

As with time-variant signals, one can observe that there are many characteristics of the signal that can be used to distinguish different types of motions or classes. Feature selection is the process of achieving this distinction by computing features of the signal while simultaneously reducing the complexity of the signal itself (Farina et al., 2010). The features primarily used in sEMG are temporal and spectral features. As a result, each feature vector generated from the window is created by concatenating the values observed in these two domains. While these time and frequency domain features can be calculated from the raw EMG data gathered, note that it is possible to also utilize raw time and raw frequency values as features as well.

Types of Features

Time Domain	**Mean Absolute Value (MAV)**	The average of absolute value of EMG signal amplitude.
	Zero Crossings (ZC)	The number of times a zero-centered signal crosses zero.
	Root Mean Square (RMS)	Used to relate contractions with rest.
	Waveform Length (WL)	Measures the effective length of the signal in the window.
	Variance (VAR)	Tries to capture the power of the EMG signal.
Frequency Domain	**Median Frequency (MDF)**	This is the frequency where spectrum is split into regions with equal amplitude.
	Mean Frequency (MNF)	This is the average frequency of the input features.
	Mean Frequency (MNF)	The frequency at which maximum power is observed.

Table 1.1: A selection of features that are used in the time and frequency domain.

1.4.2.1 Time-Domain Features

Time domain features are extracted directly using raw EMG signal allowing these features to be calculated without any transformation. These features provide temporal context of the signal in metrics defined in Table 1.1 (Phinyomark et al., 2010). The 5 time-domain features in the table above are referred to as TD5 features and are commonly used in order to characterize signals. These features are known for performing classification well at low noise environments (Phinyomark, Phukpattaranont, and Limsakul, 2012).

1.4.2.2 Frequency Features

While the raw EMG signal captures temporal information, spectral information can be obtained by performing Fast Fourier Transform on the windowed data. The resolution of frequency data can be modulated based on the FFT length. Frequency features are used often in addition to time domain features, and are often used with the goal to study muscle fatigue (Phinyomark, Phukpattaranont, and Limsakul, 2012).

It is important to note that both types of features need not be considered exclusively from one another when creating a feature vector. Instead, it is possible to train a classifier to incorporate both types of features to greatly increase signal density. One of the drawbacks from having too many features is a resultant increase in computation time thus reducing the real-time ability of the system.

1.4.3 Classifier Training

In order for a classifier to generate predictions, it needs to be trained using feature extracted EMG data acquired from a subject physically attempting to perform the requested class. The set of feature vectors associated to a specific class are known as the class dictionary and are the contributing members used when predicting the class label for an input. This data set is used as a basis of reference, and depending on the classifier, to provide a means of either generating boundaries or aid in reconstruction for each class (Scheme and Englehart, 2011). A newly initialized classifier needs data acquired on a subject by subject basis. A new user would be required to complete a training protocol in order to obtain user-specific information of each class for signal interpretation. In order to create an effective training dictionary, a classifier that utilizes hyperplanes for predictions would ideally require substantial class separation between data and reliable sEMG data in order to make more confident predictions. Such a system with high class separation would provide high performance with even the simplest classifier; however, such an ideal is rarely reached and instead there exists a lot of overlap between motions. Intuitively, certain motions such as a hand-close and finger-point lie within the same degree of freedom often resulting in the overlap of data. Class separation can be modulated to an extent based on how a subject performs the intended class - using greater contraction or by mapping certain muscular activation to a very specific class. This variation can result in the user making use of different muscles at different contractions - making it possible to separate the data.

1.5 Functional Pattern Recognition Issues

An interesting dichotomy in pattern recognition is the difference between offline performance and the performance observed in real-world scenarios (Ortiz-Catalan et al., 2015). There are many reported control schemes that show classifiers achieving greater than 90%

accuracy, but then fail to perform at that level in real-world use (Jiang et al., 2012). The primary cause for the classifier failure is that the conditions by which a subject performs the experiment do not reflect their daily conditions. Differences in the environment are not captured during the user's training and a resultant decrease in the classifier performance is observed; the untrained combinations can lead to the misclassification (He et al., 2015). In addition to changes in the environment, the inability to classify the signal could also be attributed to changes in the sEMG signals themselves. The source of these changes are henceforth referred to as either asymmetric variations or non-stationarities and are observed by changes in the following: electrode shift, limb-position effect, load effect, and muscle fatigue.

1.5.1 Electrode Changes

Electrode sites target specific muscles and are required to maintain relative orientation with each other. Unfortunately, in real-world situations, the electrodes from a prosthesis can shift due to daily activities and result in changes to the relative position of the electrodes as well the targeted muscle (Young, Hargrove, and Kuiken, 2011). This can result in a serious decline in the classification performance of the pattern recognition system. Prosthesis sockets often have sEMG electrodes oriented in fixed positions and can be easily disturbed by shifting the socket itself. This sensitivity to such stimuli makes electrode shift very pervasive (Jiang et al., 2012).

In addition to shift in electrodes, the contact between electrode and skin can be changed due to the effects of perspiration. The change in impedance can alter the recorded sEMG signal itself and result in large enough deviations resulting in misclassification. Unsurprisingly, the change in the electrode-skin contact can alter the relative position of the electrodes (Cornish, Thomas, and Ward, 1998).

1.5.2 Limb-Position Effect

The limb-position effect, as the name implies, is the failure of a classifier to predict the intended class due to a change in the limb position. Changes in limb-position can be qualitatively explained by the different muscles that need to be recruited while being quantitatively described by the different dynamic loads and torques involved at holding various positions across the elbow and shoulder joint angles (Fougner et al., 2011). Capturing all of the possible limb-positions in the target space during training is not feasible but by capturing a greater set of training positions, the detrimental effects can be mitigated to an extent.

1.5.3 Load Effect

The load effect, as the name implies, is caused by changes to the load placed on the prosthesis. Users would experience difficulty in tasks that require grasping and moving heavy objects. Studies have shown that static loads decrease classification accuracy leading to involuntary control commands (Cipriani et al., 2011). Similar to the other effects, the load effect results in changes to the characteristic of the sEMG signal that is not captured during training and can be mitigated by training with various loading conditions.

1.5.4 Muscle Fatigue

Muscle fatigue follows naturally from using a prosthesis during the day. The weight of the prosthesis and the discontinuity of load between the limb and the arm can directly implicate the user's experience during long hours, namely in the form of discomfort and muscle strain from the remaining limb. This can lead to a decrease in the classification accuracy progressively throughout the day. Interestingly, Fougner et al. (2011) in a limb-position study suggested that while muscle fatigue may be present alongside other conditions, it may not be the dominant factor for the series of misclassifications.

1.6 Prosthesis Adoption

Depending on the classifier and the degree to which the asymmetric conditions have modified the signal, prosthesis users often have no choice but to completely retrain their pattern recognition system. This would entail users undergoing the training protocol, possibly multiple times a day, to make the classifier reflect the user's conditions or account for changes in asymmetric conditions. This process can become frustrating for users due to the considerable amount of time invested in the training protocol potentially leading users to avoid using the prosthesis (Biddiss and Chau, 2007b). Due to the sensitive nature of the myoelectric systems coupled with a learning curve to effectively use the prosthesis, and the need for retraining upon failure, the needs of amputees are not often satisfied. In fact, certain professions are less likely to use a specific type of prosthesis on the basis of reliability as is the case for amputees who avoid myoelectric prosthesis if the environment requires physical labor. Such user-specific requirements place emphasis on comfort, function, and durability. Inevitably, these factors are intertwined to this dissatisfaction leads to a lower adoption rate of prosthesis and an increase in abandonment (Biddiss and Chau, 2007a).

Pattern recognition also provides real-time robust control of movements requiring several degrees of freedom, using limited information. This leads to unnatural prosthesis control and places a cognitive burden on the user in order to work with the prosthesis (Cordella et al., 2016). As previously stated, the long training sessions and limitations in clinical translatability of the developed systems from laboratory conditions makes for an unpleasant experience for users.

1.7 Pattern Recognition Updating

From the previously discussed material on asymmetric variations, it should be apparent that these are time-dependent factors that change over the course of using a prosthesis. A user therefore, will experience a decrease in classifier performance over time. This phenomenon has been observed in several studies including one by He et al. (2013), which showed that the average classification error increased over time by nearly 15% on the first day and up to 40% by the twelfth day. This phenomenon was observed across all features, which should be intuitive considering feature extraction is performed on raw data. This can be attributed to the long-term effects of asymmetric variations on the EMG data experienced by the user. The study also observed that changes *between* days were greater factors in influencing the performance degradation than compared to those *within* a day.

While literature involved in pattern recognition control scheme is often used to specifically target certain asymmetric conditions, the ultimate goal is to reduce the need for classifier retraining and assist naive users in using a prosthesis. While standalone classifiers are used as a means of generating predictions, when augmented with a signal adapting framework, they can provide a more clinically viable system for reducing the time and cognitive burden on the user (Sensinger, Lock, and Kuiken, 2009). In congruence with the two main types of classifiers, there are two different types of updating schemes. Supervised updating schemes require some way to label the input while unsupervised algorithms have to use a different set of metrics in order to make an appropriate update. Sensinger, Lock, and Kuiken (2009), found significant improvements utilizing the supervised updating scheme compared to the unsupervised approach and suggested to incorporate supervised adaptations into clinically viable systems.

Thus, with an updating method suitable for the chosen classifier, the pattern recognition performance can be significantly improved than a system without an adapting mechanism. This would not only allow the system to become robust in the presence of asymmetric variations, but also assist naive pattern recognition users in improving their control while also reducing the frustration and impairment users experience while using a prosthesis.

Chapter 2

Previous Works

2.1 Introduction

It comes without surprise that the intuitive notion of improvement in classification performance would be correlated to improved myoelectric control. For improved myoelectric control, two main strategies have been utilized: modulating the response of a prosthesis to classifier predictions (Simon et al., 2011; Scheme, Hudgins, and Englehart, 2013) and the augmentation of adaptive methods for improved classification (Chen, Zhang, and Zhu, 2013; Ams et al., 2014; Vidovic et al., 2016; Zhu et al., 2017). Unlike the latter strategy, the former does not explicitly try to improve the classifier itself. Instead, it seeks to provide the user greater prosthesis utility by adopting a different control strategy.

2.2 Control Improvement

The general philosophy of pattern recognition is that a prosthesis movement occurs because of a person's intent. Unfortunately, due the noise of a system, classifiers may predict unintended motions leading to twitchy hand motion. This severely decreases an amputee's ability to manipulate their environment. In the presence of non-stationarities, the effect of diminished control is only exacerbated. Instead of directly modifying the prediction capabilities of the system itself, an alternative control approach involves modifying the behavior of the prosthesis after a prediction has been made by first identifying if an observed signal is intentional based on some criterion. If the criterion is satisfied, then the prosthesis gets activated, otherwise no activation occurs.

Bridging off of this framework, Simon et al. (2011) previously showed that in order to minimize the effects of unintended movements, a decision based velocity ramp can be used as a post-processing step that does not interfere with the decision stream of the classifier with no additional control delay. The only modification is that the system works under proportional control which is used to modulate the speed of the intended motion. In proportional control the strength of the contraction is proportional to the speed of the prosthesis motion. By combining proportional control with velocity ramp, a grip is performed at an attenuated speed which can eventually match the maximum speed only after a series of consistent classifications.

Each class is given an ramp output speed $RG_i = \frac{C_i}{L}$, modulated by a user defined ramp gain that is dependent on some incremented counter, C_i, for some class i over a defined user ramp length L. The system also incorporated a measure of the intended activation by EMG activity in the form of average Mean Absolute Value across all N channels modulated by some boost factor B_i for class i in $V_{in} = B_i(\frac{1}{N}\sum_{k=1}^{N} MAV_k)$. The boost factor was set for each class such that normal force activation resulted in a speed equal to half of the maximum speed that could be performed by the motor. Together, these elements are used in order to modulate the activation of the prosthesis by $V_{out} = RG_i * V_{in}$. The undisturbed classifier would stream the resultant predictions but the effective control is attenuated using this velocity profile system. For a predicted class i, the counter C_i is incremented by 1 while decreasing all other classes by 2. If the stream of predictions for class i is consistent, then maximum speed could be achieved as the ramp gain is changed accordingly. If a class has reached the maximum ramp gain, the user can then control the speed of the action purely based on the force of the contraction. If however, a different class j is predicted, then the ramp gain for class i would decrease via the counter, and the ramp gain for class j would increase if consistently predicted. If a subject wanted to switch to class i from this new state, the system would attenuate the speed based on the ramp gain of class i at that time.

The study by (Simon et al., 2011) pointed out that this system provided greater control in Target Achievement Control (TAC) tests (a virtual hand movement test) and an experiment involving stacking blocks using a physical prosthesis. The results showed an increase in the completion rate, path efficiency, and completion times in the TAC experiment. In the case of the block-stacking experiment, Simon et al. (2011) monitored the total number of blocks stacked, the number of blocks that were dropped, and the tallest towers made by subjects. In congruence with the results from the TAC test, the velocity-ramp system provided better measurements across all metrics: more blocks stacked, less blocks dropped, and taller towers. The group additionally reported that users verbally expressed "less frustration and finer positioning for small adjustments while using the velocity ramp" while some wished for greater control when using proportional control (Simon et al., 2011). This velocity ramp based system ultimately provided better control of the prosthesis compared to a system with a majority vote or a system without an output filter.

Another approach explored in (Scheme, Hudgins, and Englehart, 2013) coupled a confidence score for each decision generated by an LDA classifier. If the projected confidence score is above or below a particular threshold, the system can choose to either execute the decision or instead default to a no motion class. An LDA generated prediction is based on the class that maximizes the log probability of a class k that produces input x given by $\ln(P(\omega_k|x))$. This returns the prediction of the classifier, but the associated confidence can be obtained by the softmax of $\ln(P(\omega_k|x))$ over all classes K. Hence, a confidence score for an input x is given by $C_k(x) = \frac{e^{\ln(P(\omega_k|x))}}{\sum_{k=1}^{K} e^{\ln(P(\omega_k|x))}}$. As (Scheme, Hudgins, and Englehart,

2013) describe, this confidence function provides a value between [0, 1] to the associated prediction for class k. If the confidence $C_k(x)$ is greater than a confidence threshold $R_{LDA,k}$, then the predicted class is returned, otherwise, the rest class or no motion class is returned. Like (Simon et al., 2011), this system's framework or producing a confidence score is also dependent on proportional control. The system that incorporated this active rejection mechanism to LDA is referred to as LDAR.

This rejection based approach by Scheme, Hudgins, and Englehart (2013) was tested offline by measuring classification accuracy and Fitt's parameters by performing a pseudo 3D Fitt's Law test. In the first offline experiment, the subjects were required to perform a series of motions over 2 seconds in ramp like contractions; gradually increasing the intensity of contractions until a moderate level is achieved. In the Fitt's Law Task, the subject was required to move a cursor (horizontally, vertically, and radially expand/contract) into a final target zone. The final target zones are different in distance and size for varying levels of difficulty. A test is deemed successful if within the 15 seconds of time given to complete the task, the subject was able to keep their cursor located within the target location for 1 second. In the offline analysis, a confidence rejection threshold of 0.97 was used. Classifier decisions less than 0.97 confidence were rejected and relabeled as no movement. The high rejection score led to few incorrect active decisions and some false rejections during the active phases of subject motion. The Fitt's Law Test illustrated that the active rejection system produced significantly better results in all metrics compared to the base system for both able-bodied and amputee subjects. An improvement of 43% and 47% was observed for throughput, 21% and 29% higher path efficiency, 69% and 58% lower overshoot, 36% and 23% decrease in stopping distance, and 7% and 10% increase in completion rate for able-bodies and amputees respectively. The increase in throughput would suggest that the LDAR system does improves prosthesis controllability for amputees, while simultaneously rejecting incorrect classifier predictions. Additionally, since confidence scores are dependent on proportional control, low proportional control amplitudes exhibited increased rejection. Like in (Scheme and Englehart, 2011), this method of influencing the impact of the classification prediction shows improvement to how well a subject can control a prosthesis. By making confidence rejections, the system acts as another check to increase the likelihood that a classifier prediction matches the subject's intended movement.

Another classification system dependent on confidence rejection was proposed in (Ams et al., 2014). In contrast to (Scheme, Hudgins, and Englehart, 2013) where a confidence score was provided based on the probability of the prediction, Ams et al. (2014) trained an artificial neural network (ANN) on the contraction level and prior classifier predictions in order to remove misclassifications. Instead of using the MAV across all channels to calculate the force of the contraction like in (Simon et al., 2011), RMS was used instead. First, a feature extracted signal was sent to LDA where a prediction was made while simultaneously calculating the average muscle activity. The LDA predictions and the RMS values of the input at some time *t* to *t*-10 are fed into the ANN where it generated a confidence prediction. As a new prediction is made every 50ms, this would imply that the last 500ms worth of data is used in order to generate a confidence prediction. The output of the ANN is either a +1 if correct, or a -1 if incorrect. Using this information, a Trust Index (TI) value was subsequently calculated; the TI is what is ultimately used to either accept or reject the prediction and is given by $TI(t) = |TI(t-1)|^{\alpha n(t)} + \beta(t)$. Here, α represents a smoothing factor, $n(t)$ represents the output of the ANN, TI(t-1) represents the trust index of the previous decision and β(t) is an integration factor that reflects the increasing confidence of a decision. All of these factors are used in order to calculate the TI(t). If TI(t) is greater than some threshold θ_{TI}, then the class label by LDA is used. If TI(t) is less than θ_{TI}, then the class label by LDA is rejected and the previous class is maintained.

To test this ANN confidence rejection method, Ams et al. (2014), created two models using the ANN system: one with globally optimized parameters for α and θ_{TI} and the other with subject optimized (ANN-IND) α and θ_{TI}. They compared their systems with base LDA, LDA with a majority vote (LDA-MV) with a length 9, the model proposed by Scheme, Hudgins, and Englehart (2013) (LDA-RJNM), and one variation of the previous model but by defaulting to the last known prediction instead of no motion (LDA-RJRM). Equipped with eight differential electrodes, subjects were required to have elbows flexed at an 90deg angle while resting their arm on the backrest of the seat and were then instructed to perform eight different movement classes and vary their contraction strength to follow a trapezoidal strength profile with a 1s rise, 3s plateau, and 1s fall. Each motion was repeated 5 times at 3 different plateau contraction levels: 30%, 60%, and 90% of the maximum contraction force.

	LDA-MV		**LDA_RJNM**		**LDA-RJRM**		**ANN-GO**		**ANN-IND**	
	tAcc	aAcc	tAcc	aAcc	tAcc	aAcc	tAcc	aAcc	tAcc	aAcc
LDA	2,44	3,28	**-25,64**	**20,38**	-2,22	**11,05**	**4,60**	**21,58**	**5,92**	**31,51**
LDA-MV			**-28,09**	**17,10**	**-4,66**	**7,77**	2,16	**18,30**	3,48	**28,23**
LDA_RJNM					**23,43**	**-9,33**	**30,25**	1,20	**31,57**	**11,13**
LDA-RJRM							**6,82**	**10,52**	**8,14**	**20,46**
ANN-GO									1,32	**9,93**

Figure 2.1: A table taken from (Ams et al., 2014), that captures the difference in accuracies between classifiers. For each cell, a positive value means the classifier in the column outperforms the classifier in the row. A negative value implies the classifier in the row outperforms the classifier in the column. The bold values indicate significance for $p<0.05$. **LDA-MV** is LDA with a majority vote of length 9. **LDA-RJNM** is the rejection scheme that treats the rejected class as no motion. **LDA-RJRM** is the rejection scheme that relabels the rejected prediction to the last accepted prediction. **ANN-GO** is the rejection scheme that uses a trust index with globally optimized values. **ANN-IND** is the rejection scheme that uses a trust index with individually optimized values.

Two measures of accuracy were presented: $tAcc = \frac{CorrectClassifications}{TotalClassifications} * 100$ - which takes rest into account - and $aAcc = \frac{CorrectActiveClassifications}{TotalActiveClassifications} * 100$ - which does not take rest into account (Ams et al., 2014). This was done to separate when a misclassification occurs due to an active motion versus rest. With an $\alpha = 0.2$ and $\theta_{TI} = 0.61$, ANN-GO and ANN-IND outperform base LDA by 21% and 32% in aAcc and tAcc respectively. In comparison to LDA-RJNM, ANN-GO and ANN-IND outperformed in tAcc by 30%, but ANN-GO was only marginally better (1.2%), in aAcc, while ANN-IND saw an 11% increase. Since both tAcc and aAcc provide a different interpretation of the "correct" classification, results from both were considered together. Together, these results suggest that the use of historical data in generating a confidence prediction has *potential* in providing improved prosthesis.

In all of these proposed methods, there was no change to the inherent prediction ability of the system, rather, the focus was on identifying if an input was due to noise or user intent and subsequently modulating the prosthesis dynamics. This implies that misclassifications, while acceptable in these frameworks, are diminished by either defaulting to a no-motion class, changing movement speed, or by defaulting to previously predicted class. While these systems do not directly address the issues of the temporal effects of non-stationarities, they propose solutions that have some means of controlling the effects of misclassifications for greater prosthesis control.

2.3 Adaptation Methods

In contrast to the previous systems, several groups have tried to directly modify the way a classifier provides a prediction by using an updating system to modify a classifier's base parameters. After an initial training session, a classifier is instantiated using the acquired training data to optimize its parameters for future predictions. In the case of

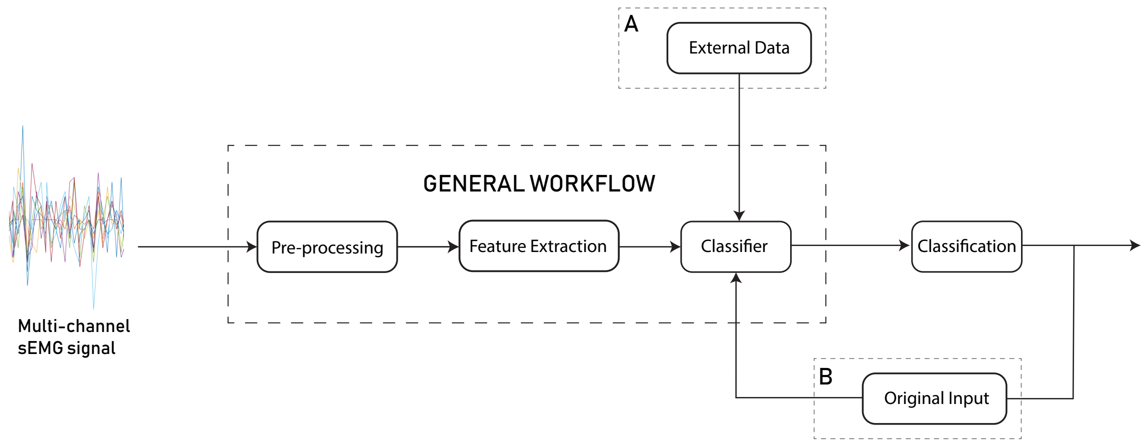

Figure 2.2: A general image illustrating the work-flow of information in previously explored systems. (A) represents the use of external data, like a calibration set, in order to tune parameters of the classifier. (B) represents the use of the original input itself, as employed by the self-enhancing system, in order to update the classifier. Both ways are useful for improving classification performance.

boundary classifiers, the training data is effectively compressed into parameters that define class boundaries, thereby relieving the need for data storage. Other classifiers, may require active dictionary maintenance. Due to the effects of non-stationarities, the data used to initialize a classifier is no longer reflective of the environmental conditions faced by the subject. In the case of LDA, an updating system would have to modify the parameters of the class means and pooled class covariances in order to reflect the user's environment. If instead of LDA, QDA is used, then the covariance for the appropriate class is also updated. A general overview of this process can be visualized in Figure 2.2. By modifying specific parameters, the updating scheme is likely optimized for a classifier that make use of complementary values. In (Chen, Zhang, and Zhu, 2013), the LDA classifier's parameters are continuously updated as the input streams. This self-enhancing method, as coined in Chen, Zhang, and Zhu (2013), takes the feature extracted EMG sample alongside its classifier prediction and uses them to update the parameters for class mean and the covariance matrix. This implies that the information from the original training data is retained and added to by newly streaming testing data. This allows the contributions of the training data to become more prevalent as the amount of testing data incorporated will exceed the training data for that class over time. Additionally, this method allows an asymmetric amount of information to be maintained per class as updates occur.

Subjects were required to perform 10 motions with their arm relaxed towards the ground while performing each motion with a natural force for 5 seconds after 5 seconds of rest. All data was collected using 4 channels of sEMG electrodes with a band-pass of 8-500 Hz at a 1000 Hz sampling frequency. The data was collected for 20 cycles, where each cycle required subjects to perform all 10 classes. All analysis was accomplished offline after acquiring data in two different ways: first, collecting testing data immediately after training, and second, by collecting testing data 6-7 hours after collecting the training data. In the first protocol, 20 cycles of data was acquired whereby the first 6 cycles were used as a training data set, and the last 14 were used as testing sets. In the second training protocol 35 cycles worth of data was collected where the first 5 cycles of data was used for training while the last 30 were used for testing. All signals are feature extracted using AR + FC (Fourier-derived Cepstral Coefficients) (Chen, Zhu, and Zhang, 2009) for their improved performance compared to other EMG features (Chen, Zhang, and Zhu, 2013).

The above self-enhancing system was not only able to provide higher classification performance, but also had a smaller standard deviation than the base classifier in performance (Chen, Zhang, and Zhu, 2013). Comparisons were made using LDA and QDA with their respective self-enhancing methods SELDA and SEQDA using AR or FC features (Chen, Zhang, and Zhu, 2013). Quadratic Discriminant Analysis (QDA), is a variant of LDA, whereby each class is given a covariance instead of a single pooled covariance.

Accuracies of Classifiers With Different Features				
	LDA	**SELDA**	**QDA**	**SEQDA**
AR	93.57±2.54	95.11±2.25	93.13±4.24	95.34±3.32
FC	94.08±3.26	95.67±2.52	95.47±2.28	97.62±1.87

Table 2.1: Accuracies (given as %) of the base classifier and the self-enhancing classifier using either AR or FC features. The values are presented in mean and standard deviation format and are all interpreted as percents. This table simply compiles some information from (Chen, Zhang, and Zhu, 2013).

When comparing the results from the second protocol, SEQDA and QDA performed better than SELDA and LDA and was hence the focus of discussion (refer to Table 2.1). In general, the updating system performed significantly better than the base counterparts ($p < 0.01$) when assessed by a paired t-test. Covariance updating did not greatly improve performance for SELDA, but was important for SEQDA. This makes sense considering QDA assumes that each class has its own covariance, thus, class-wise covariance updating can be very helpful in QDA. Since SEQDA performed much better with a covariance update, it is expected that the "second order information" of class-covariance updating is better than pooled covariance. These findings suggest that updating parameters that best suit the respective classifier is perhaps the ideal method.

Instead of using streaming data to gradually change the means and covariances of classes, another proposed method sought to utilize a small calibration set to modify the means and covariances. The model adaptation is based on modifying the mean and covariance matrix of the classifier by some weighted contributions of the calibration data set's class means and covariance matrix and the classifier's class means and covariance matrix parameters. A new class mean is therefore calculated like so: $\mu_{c,new} = (1-\tau)\mu_{c,tr} + \tau\mu_{c,cal}$ and $\sigma_{c,new} = (1-\lambda)\sigma_{c,tr} + \lambda\sigma_{c,cal}$. Let c represent the class of interest, and τ and λ represent some regularization parameter that is obtained via a grid search. Effectively, the system tries to take data more reflective of the environment the user is more likely to face, and places more weight towards the class means and pooled covariance observed in the calibration set while simultaneously trying to hold some amount of information from the training data class means and covariance. This approach is similar to that of Chen, Zhang, and Zhu (2013); the LDA classifier is updated by means and covariance, the two parameters that are required for optimum classification.

Over a period of 5 days, subjects were requested to perform 8 motion classes while seated in a chair and bending their arm in a 90deg angle. All movements were recorded at three different contraction forces of 30%, 60%, and 90% of the maximum contraction force while the subject was equipped with 8 electrodes and hard socket prosthesis. Data was collected in a similar manner to (Ams et al., 2014), where users were requested to perform 5 runs of all 8 classes at the 3 contraction levels. As a result, a total of 120 trials were obtained per session where each trial required users to perform a motion at varying contraction for 5 seconds: 1s rise, 3s plateau, 1s fall. Over the course of the 5 days, the first session of the first day was used as a training set, while the first session of all the other days were used for testing. The second session of each day was used as validation for further parameter optimization using the proposed method. All data was acquired by streaming at 1000 Hz, and filtered between 20-450 Hz with an additional notch filter at 50 Hz. A value of 0.8 was used for both τ and λ. Several types of LDA updating systems were introduced in addition to the base classifier as given in Table 2.2.

Different LDA Models Tested	
LDAMA	LDA's mean is updated
LDACMA	LDA's mean and covariance is updated
LDAnew	LDA trained on calibration data

Table 2.2: Various models of LDA used in (Ams et al., 2014). Each model accomplishes the same thing, either update the mean or the covariance. The only difference lies in the amount and the data sets used for the update.

After preliminary analysis on performance of various LDA methods, Vidovic et al.

(2016) focused on base LDA, LDAnew, LDAMA, and LDACMA. LDAMA performance remained consistently above 90% and illustrated similar results to LDACMA. Compared to the base classifier, LDAMA and LDACMA significantly outperform LDA, but LDACMA performed significantly better than LDAMA in amputees. They observed that LDACMA performed better for 70.4% of subjects compared to LDAnew. They also report that subjects who used LDACMA percieved the controllability better than LDAnew, even though the former was exhibiting decreasing performance, while the latter was showing increased performance. This same analysis was accomplished for QDA and observed that the covariance adaptation provided more improvement than for LDA. Overall the authors thought that the approach provided a reasonable tradeoff between high classification accuracy over days and minimal user retraining, as obtained by the calibration set. These results suggest that the use of a calibration set as a means of optimizing the classifier's parameters at discrete times provides significant improvement to the base classifier.

A recent paper (Zhu et al., 2017), illustrated a system that effectively combined the previous two systems. The core idea is to have a component of the adaptive system that utilizes a small calibration set like in (Vidovic et al., 2016) to calibrate for inter-day changes while having a self-enhancing system for intra-day changes as observed in (Chen, Zhang, and Zhu, 2013). Just like in Vidovic et al. (2016), where τ and λ are used as regularization parameters, Zhu et al. (2017) adopt r as a regularization parameter (or learning rate as coined) used to update the class-wise means and pooled covariance. Then an additional self-enhancing aspect is further augmented in-tandem with the exact same mechanism as described in (Chen, Zhang, and Zhu, 2013). This way, the domain adaptation is used to reduce training samples in the training stage while the self-enhancing framework is able to compensate for the slowly modifying sEMG signals experienced from non-stationarties.

	Classification Methods			
	LDA-BL	LDA-SE	LDA-DA	LDA-CA
	Intact-limbed Subjects			
Mean±Std	88.24±5.80	92.80±5.27	91.29±4.82	94.99±3.97
Groups	C	B	B	A
	Amputee Subjects			
Mean±Std	79.94±6.97	85.05±5.90	83.81±7.47	88.67±4.42
Groups	C	B	B	A

Figure 2.3: Table taken from (Zhu et al., 2017) illustrates the classification accuracy using different LDA updating methods. The subject pool includes different groups of amputees and able-bodied subjects.

Data for able-bodied subjects was acquired using 4 sEMG electrodes while amputee subjects were given 6 sEMG electrodes. Subjects were required to perform 13 different motions if able-bodied, and 11 if amputee, where each motion was sustained for 5 seconds followed by 5 seconds of rest. From each subject, 20 trials of data was captured per session and was collected over 10 days. Data was bandpass filtered between 20-450 Hz, and sampled at 2 kHz. 6th order auto-regressive (AR) features were used alongside 4 time domain (TD) features: mean absolute value, zero crossings, waveform length, and slope sign changes. The classifier was trained using data acquired on the first day; the first 10 trials were used for training and then evaluated using the last 10 trials. For days 2 - 10, the first trial is used for domain adaptation as the calibration set, while the subsequent 19 trials are used for self-enhancing. An online test was also performed, where 10 trials of data was acquired for 11 motions. Each motion was performed for 5 seconds followed by 5 seconds of break with real-time visualization. One testing session was performed immediately after the training session and another testing session after 2 hours. On the second day, 1 training set was acquired alongside 3 testing sets, where the first and second testing set are 2 hours apart and the second and third are 1 hour apart. Each class was given 5 seconds to perform, and only required 10 correct classifications within the specified time. Subjects were actively provided visual feedback in the form of a virtual prosthesis.

Performance Metric		LDA-BL	LDA-SE	LDA-CA
MST (s)	Mean±Std	0.42±0.14	0.40±0.08	0.36±0.06
	Groups	A	A&B	B
MCT (s)	Mean±Std	1.45±0.16	1.35±0.10	1.29±0.07
	Groups	A	B	C
MCR (%)	Mean±Std	97.13±3.36	97.55±3.23	98.68±1.95
	Groups	B	B	A
RTA (%)	Mean±Std	81.76±6.05	87.29±5.17	90.07±3.28
	Groups	C	B	A

MST = Motion Selection Time; MCT = Motion Completion Time; MCR = Motion Completion Rate; RTA = Real-Time Accuracy.

Figure 2.4: Table taken from (Zhu et al., 2017) that illustrates the performace of subjects in different groups performing the online experiment. All of the performance metrics given are metrics for task completion.

The Cascaded Adaptive framework (LDA-CA) system outperformed the base classifier while also keeping a small standard deviation in terms of rote classification accuracy implying greater consistency while achieving strong classification in offline analysis (refer to Figure 2.3). Additionally, in the online experiment, Zhu et al. (2017), the LDA-CA model outperforms the base LDA in the online experiment with better average values and with greater consistency in all of the previous metrics (refer to Figure 2.4). This suggests that with respect to mean performance, the LDA-CA model performs better overall and is more consistent, as given by the smaller standard deviations across all metrics.

It should be noted that the value of the learning rate r implies that changes in mean and covariance should be heavily weighted towards the calibration set over the training set. As a result, improvement is to be expected as the system places more weight towards input from the new environment. In addition to the offline analysis, the online analysis suggested that with real-time feedback to the user, the proposed method improved performance within a day for some subjects. This supports the notion that in an online experiment, real-time feedback can influence subject behavior and thereby affect performance. Consequently, offline results are not always bound to be consistent with the online results from experimentation. For example, the proposed Cascaded Adaption framework (DA + SE) did worse than the base LDA when the base LDA was completely retrained in offline data. In the online experiment, LDA-CA does just as well, if not, better than the full training in the first day; thus, suggesting that an online experiment is an important factor for understanding myoelectric control. Overall, the Cascaded Adaptive framework was able to improve the classification ability by using a calibration set and then continuously updating the model parameters from all further streaming data.

The above systems illustrate LDA requires a lot of data in order to be a robust classification system. Furthermore, by choosing LDA, there is an inherent assumption that the data distribution for each class is gaussian which is not necessarily true. Perhaps the most interesting aspect of the above systems is that for the Domain Adaptation framework originally observed in Vidovic et al. (2016), and later in Zhu et al. (2017), set the value of the learning rate or regularization parameter close to 1. This implies that each Domain Adaptation update places a greater emphasis on the values from the calibration set, even though the number of samples from the calibration set at the point of the update are much smaller than all of the samples used previously to contribute to fine-tuning the classifier's parameters. From these previous works, there is strong support in suggesting improved classification performance when augmented with an updating system of some kind.

2.4 Conclusion

While all of the previous methods assume temporal variation of the sEMG signal and optimize classifier's parameters by collecting all of the streaming input or utilizing a calibration set, we propose an alternative selection criterion based on an different classifier and an updating method that takes advantage of signal structure. Instead of taking vast amounts of data, we will optimize on a smaller data set while selectively choosing members to incorporate and replace our classifier model. From the above systems, there are several key takeaways:

1. The same methods used to optimize one classifier will not necessarily hold when using a different classifier.
2. Using small chunks of data, like a calibration set, is a useful method for updating a classifier.
3. Keeping historical data is important in maintaining classification accuracy.
4. While offline analysis is useful for tuning the updating system, the performance may not be the same when tested real-time with subjects.
5. Updating system augmented classifier outperform their base classifier counterparts in improving rote classification accuracy.

These ideas are considered and incorporated in the updating method explored in the next chapter.

Chapter 3

Information Centric Retrospective Updating with EASRC

3.1 Introduction

Pattern recognition - for all of its potential - has challenges with clinical robustness. One of the major functional challenges in prosthesis use stems from asymmetric variations: changes in electrode positions, limb positions, force, and electrode conductance. This breakdown generally occurs because the classifier is trained in an environment that does not reflect the testing conditions. As a result, the model used to generate the prediction fails as the new input does not resemble the intended motion; however, it may fall within a region of the model defined by a completely different motion leading to a misclassification. This performance decline can often cascade and result in poor controllability, at which point a user would need to completely retrain their pattern recognition systems by undergoing a training protocol to account for their new conditions. Unfortunately, this not only an arduous task to repeat, but also an intractable approach for a user. As many of these asymmetric variations are transient changes, there exist a need for an updating system that is capable of maintaining a model more reflective of the user's immediate environment. EASRC and the associated updating system were implemented in MATLAB R2018a.

3.2 Introduction to EASRC

3.2.1 Overview of EASRC

EASRC is the supervised classifier of choice in this updating system because of its intrinsic robustness to asymmetric variations (Betthauser et al., 2018). The reason for this has more to do with the hybrid system that comprises EASRC: a combination of an Extreme Learning Machine (ELM) and Sparse Representation Classification (SRC). Contrary to its name, the ELM is simply a single hidden layered neural network, where each node has randomly generated input weights and biases. The only parameters being optimized in an ELM are the output weights from each neuron. The idea is that each neuron (or node) generates a hyperplane across the training data to generate boundaries that are used to demarcate each class in a supervised problem space. This simplistic architecture makes ELMs computationally fast, due to the decreased number of parameters to optimize for, but is not robust to noisy inputs like non-stationarities. SRC on the other hand, is robust to noise but is computationally slow. Contrary to the behavior of ELMs and boundary based classifiers, Sparse representation classification isolates the class of interest by first finding the minimized l1 solution that best captures the input from a combination of the training dictionary samples thereby providing the *sparsity coefficients* of each contributing vector. The contributions of each class participating in the minimized l1 solution is measured by taking the contributing vectors from each class exclusively and their respective sparsity coefficients (weights) to reconstruct the input vector. The class that minimizes the resultant residual is chosen as the prediction; effectively, the class that contributes the most to the reconstruction of the input is returned. This approach allows SRC to not rely on the inherent need for class separation in the feature space in order to provide an accurate prediction - making it a robust classifier.

EASRC's hybrid system takes advantage of the ELM's speed and SRC's robustness in order to quickly and effectively classify inputs. In EASRC, a test input will first parse through the ELM which returns a set of probabilities associated to each class. The two classes with the highest probabilities are then compared. If the difference between these probabilities is greater than some threshold α, then the ELM is confident in its prediction and will return the class with the highest probability. If the difference between the first and second highest probabilities are less than α, the ELM is not confident in its decision and then forwards the classification task to SRC. Instead of equipping SRC with the entire training dictionary from all K classes, the probabilities from the ELM output are used to make a reduced dictionary comprised of samples from the k most likely classes. This reduces the computation time of SRC significantly and also makes efficient use of the ELM output. While ELM looks to classify based on the boundary of classes, SRC seeks to explain an input by a class's vector space. As a result, the reconstruction of the input makes the system robust as even a single significant vector contribution within a class can greatly improve the classification ability of EASRC for that class. This behavior is reflective of EASRC's focus on information content. If the appropriate information (in the form of a feature vector) for a class is present within the training dictionary, then any new information (input feature vector) similar to the training sample can be accounted for by the weighted contribution of that training sample (see Figure 3.1).

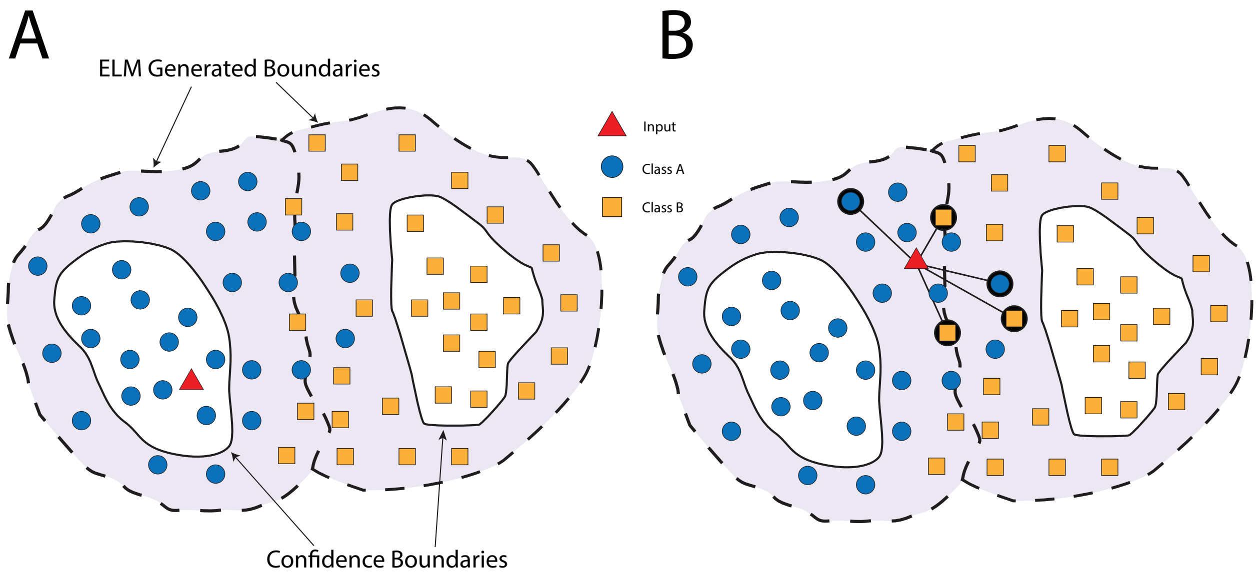

Figure 3.1: This figure illustrates the two different possibilities of EASRC. In (A) the given input is within the confidence boundary of the ELM and will hence return class A. In (B), the given input is outside the confidence boundary and as a result uses SRC to obtain the minimized L1 reconstruction. Note here, that even though the input is in the ELM boundaries for class A, if class B contributes more to the reconstruction, then class B is returned. If class A contributes more to the reconstruction, even with less vectors, class A will be returned. Significance of reconstruction matters more than the number of vectors participating in the reconstruction.

3.2.2 EASRC in EMG

As shown by Betthauser et al. (2018), EASRC shows tolerance to the limb-position effect when compared to currently utilized classifiers like LDA. Over an untrained positional variation in space, EASRC is able to maintain robust classification accuracy and experiences minimal decrease in classification performance. This was exemplified in an offline analysis, which contained data acquired for 2.5 seconds at 81 different positions from eight able-bodied subjects and two amputees while they performed five classes (rest, hand open/close, wrist pronate/supinate) at each position. An additional online experiment involving subjects performing the same five motions at five different positions on a planar board. A user would align their hand in a neutral (90deg) position and train 3 repetitions of the five movement classes at the training position to obtain a training dataset. The user would then test these five motion classes at four different positions, two of which were located 23cm to the left and right of the training position, and the other two 46cm above those left and right testing positions. A trial required the subject to perform a modified TAC test 3 times at each position; four such trials were conducted for LDA and EASRC and during the experiment, during which the subject was unaware of which classifier was running in the background. A successful TAC test was one where the subject is able to complete the a given motion within 15 seconds and hold the target position with a dwell time of 1 second.

Analysis of the acquired offline data was performed in two different manners: consider any 2 of the 81 trials and train with one and test on the other; and train using n trials and test on all (81 - n) others. By considering 2 trials at a time, the contributions of distance from training position on classifier performance can be measured and is marked with a decrease in classifier performance as the distance from training location increases. When ascribing a linear fit to the classifier performance observed by LDA and EASRC, we observe the rate of decrease with respect to distance for EASRC is nearly 54.2% less decay (-0.00236 %/cm versus -0.00108 %/cm) compared to LDA. By training on n trials and testing in all others (81 - n) trials, the performance capabilities of random multi-position training can affect testing performance. From this analysis, EASRC performed better when trained in 2 random positions than other classifiers trained in up to 6 positions. Results from the online experiment illustrated that subjects using EASRC for online control were able to complete all of the target tasks in all positions; LDA resulted in failure when attempting to perform certain tasks - especially hand close. Subjects were able to complete the "tasks faster, with less throughput, and greater throughput" when using EASRC compared to LDA ($p < 0.05$) (Betthauser et al., 2018). From this analysis, Betthauser et al. (2018) showed that EASRC provides a classification strategy that is more robust to asymmetric variations alongside minimal user training and is hence, the base classifier used in this thesis.

3.2.3 The Mathematics Behind EASRC

While EASRC is a hybrid classifier that depends on ELM and SRC, we will consider both components independently, yet within the dynamics of their hybrid nature.

3.2.3.1 ELM

As previously stated, ELM is a single-layered neural network that has randomly initialized weights and biases, but optimized output weights β. We will now consider a system with L hidden nodes, K classes, N samples, d features, and an input $\mathbf{x}_j \in \mathbb{R}^d$ which produces an output $\mathbf{o}_j \in \mathbb{R}^K$ and is compared to known output $\mathbf{t}_j \in \mathbb{R}^K$. Each node has an associated $\mathbf{w}_j \in \mathbb{R}^d$ corresponding to the weight at the node, and a corresponding bias b for each node. Thus the overall ELM equation can be represented as

$$\sum_{i=1}^{L} \beta_i g_i(\mathbf{w}_i * \mathbf{x}_j + b_i) = \mathbf{o}_j$$

In order to minimize the error of the system, ELM aims to solve the following in order to tune the output weights β

$$\min \|\mathbf{H}\beta - \mathbf{T}\|_F + \frac{1}{\lambda}\|\beta\|_F$$

Here, λ is a regularization parameter, $\mathbf{H} \in \mathbb{R}^{NxL}$, $\mathbf{T} \in \mathbb{R}^{NxK}$, $\boldsymbol{\beta} \in \mathbb{R}^{LxK}$ and are given like so

$$\boldsymbol{H} = g(\boldsymbol{x}^T\boldsymbol{w} + \boldsymbol{b}) = \begin{bmatrix} g(\boldsymbol{w}_1 \cdot \boldsymbol{x}_1 + b_1) & \dots & g(\boldsymbol{w}_L \cdot \boldsymbol{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{w}_1 \cdot \boldsymbol{x}_N + b_1) & \dots & g(\boldsymbol{w}_L \cdot \boldsymbol{x}_N + b_L) \end{bmatrix} \boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_N^T \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}$$

Before calculating the final output weights β, the optimum λ denoted as λ_{opt} is first obtained by a Leave One Out (LOO) approach. As the name implies, the LOO method takes N samples of data and makes N different training datasets, where it leaves one sample from the training (N-1 samples) and instead uses it to evaluate the training model. This method is accomplished by measuring the residual sum of squares (PRESS) statistic (Myers and Myers, 1990; Cao et al., 2016) given as:

$$MSE^{PRESS} = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\mathbf{t}_j - \mathbf{o}_j}{1 - (HAT_r)_{jj}}\right)$$

$(HAT_r)_{jj}$ is defined is the jth diagonal element in the SVD decomposed calculation of the output matrix **H**. Let **H** = $\mathbf{UDV^T}$

$$HAT_r = \begin{cases} \mathbf{HV}(\mathbf{D^2} + \breve{}\,\mathbf{I})^{-\mathbf{1}}\mathbf{V^T H^T}, & \text{if } L \leq N \\ \mathbf{HH^T U}(\mathbf{D^2} + \breve{}\,\mathbf{I})^{-\mathbf{1}}\mathbf{U^T}, & \text{if } L \geq N \end{cases}$$

Note the presence of λ in HAT_r. The λ_{opt} is determined by going through a defined range of values between λ_{min} to λ_{max} and choosing the value that minimizes MSE^{PRESS}. Once calculated, the final output weights β are calculated like so:

$$\beta = \begin{cases} \mathbf{V}(\mathbf{D^2} + \breve{}_{\mathbf{opt}}\mathbf{I})^{-\mathbf{1}}\mathbf{V^T H^T T}, & \text{if } L \leq N \\ \mathbf{H^T U}(\mathbf{D^2} + \breve{}_{\mathbf{opt}}\mathbf{I})^{-\mathbf{1}}\mathbf{U^T T}, & \text{if } L \geq N \end{cases}$$

Compared to a manually selected regularization parameter, the LOO approach provides better performance (Cao et al., 2016). By directly computing the eigendecomposition on $\mathbf{H}\mathbf{H}^T$, the computation time required to computer SVD is avoided especially when $L \geq N$.

From this initialization, the ELM performs classifications by returning the class that is associated to the largest entry from the output vector $\mathbf{o}$. In EASRC, there is a slight modification, first, the difference in values are calculated in a variable noted as $T_{diff} = \mathbf{o}_f - \mathbf{o}_s$ where $\mathbf{o}_f$ and $\mathbf{o}_s$ are the first largest and second largest entries of the output vector $\mathbf{o}$. If $T_{diff} > \alpha$, where α is some confidence threshold, then the class associated to the largest entry $\mathbf{o}_f$ is returned. Otherwise, the dictionary is reduced to k classes, as uncorrelated classes are likely to have little influence in predicting the input, and the reduced dictionary is parsed to SRC alongside the input vector.

3.2.3.2 SRC

One of the key aspects of SRC is that it requires a dictionary of stored samples for input reconstruction. This is in contrast to LDA, QDA, ELM, or ANNs where the data is useful in initializing and optimizing the required parameters, but is not necessarily stored. In SRC, an over-complete dictionary is used to represent the classes and is prone to being potentially unstable if highly redundant due to the negative effects of uncorrelated classes. If the value of $T_{diff} < \alpha$, then out of all the classes that compose the dictionary $\mathbf{A} = [A_1|A_2|A_3|\ldots|A_K]$, only the k most significant classes are chosen to make a new sub-dictionary $\mathbf{A}^* = [A_{K(1)}|A_{K(2)}|A_{K(3)}|\ldots|A_{K(k)}]$, where $A_{K(i)} \in [1, 2, 3, ...K]$. This new sub-dictionary greatly reduces the computational constraint of solving the sparsity coefficients across the global dictionary. As a result, the sub-dictionary can be used to solve the following instead:

$$\hat{\mathbf{x}} = \arg\min_{x} \|\mathbf{A}^{*}\mathbf{x} - \mathbf{y}\|_2^2 + \tau\|\mathbf{x}\|_1$$

Note that by only including the k most significant classes in the sub-dictionary, this implies that the sparsity coefficients of other classes is 0. Once $\hat{\mathbf{x}}$ is obtained, we then attempt to reconstruct the input, $\mathbf{y}$, to SRC $\mathbf{y}$ to determine which class from the reduced sub-dictionary contributed most to the reconstruction. Let $\delta_c(\hat{x})$ represent a function that considers only the non-zero samples of class c from the sparsity solution $\hat{\mathbf{x}}$. Therefore, the final output label can be calculated by finding the minimized residual.

$$Label(\mathbf{y}) = \arg\min_{c} \|\mathbf{y} - A^{*}\delta_c(\hat{\mathbf{x}})\|_2^2$$

In a way, the ELM of EASRC behaves as a filter by removing classes that are not relevant so that the robust classification from SRC can take precedence in providing a prediction.

These complementary behaviors provided by the ELM and SRC satisfies the real-time and robust classification restrictions needed in the EMG problem space. The ability to maintain an active dictionary makes it possible to selectively retain specific samples that may assist in classification prediction while removing samples that may lead to misclassifications. Together, these attributes make EASRC a solid classifier to build an updating system around.

3.3 Overview of the Updating System

While Betthauser et al. (2018), showed that EASRC is more robust to the limb-position effect than base LDA under the defined parameter constraints, it is clear that the decrease in classification accuracy still exists. In order to address this issue, we can consider the intuition behind EASRC. Namely, due to its hybrid nature, the ELM provides a probabilistic and spatial representation of a new input by predicting which class boundary the input is encapsulated within. SRC, however, tries to find the class that captures what class the input is most like. Additionally, all of the information contained in the original dictionary that the ELM uses is compressed into the randomly initialized weights and biases, as well as the optimized output weights, β, generating the class boundaries and therefore does not require the storage of the training data. However, SRC, which is only used when the prediction of the hybrid classifier is not confident, requires a stored dictionary of samples from all classes to be maintained. Hence EASRC also requires an actively maintained dictionary.

While there are several approaches in devising an updating system for EASRC, optimizing for SRC first allows the classifier to be optimized on different principles. Since SRC tries to find a class with samples representative of the input, modifying these samples corresponds to modifying contextual information (refer to Figure 3.1). Thus, by taking an input over some duration of time t and capturing the structure of the input and incorporating such samples into the dictionary, similar future inputs can be accurately predicted by SRC. This new contextually updated dictionary can then be used to recalculate new ELM output weights, β, optimizing spatial information.

3.3.1 Expected Application

The updating system is comprised of two main components: the updating method and the classifier. A system that incorporates the updating system would need to store streaming feature extracted sEMG data as a subject is using a prosthesis. This stored data represents the subject's environment from their activity at some point in time and is representative of their intended action. During a subject's intended motion the classifier may predict the incorrect class which would be visualized by the prosthesis. The subject may wish to correct the classifier and would do so by performing an update. An update would require the subject to record some amount of information acquired in the conditions that reflect the classification failure. This collected information can then be relabeled to the subject's intended class and incorporated into the respective class sub-dictionary. Subsequently, the updated dictionary is used to recalibrate the classifier's parameters to better predict the motion the next time a similar input occurs. In order for the update to be meaningful, there is a focus on information content - wanting to minimize the number of samples added to the training dictionary while maximizing the impact of each update. This flow of information can be visualized in Figure 3.2.

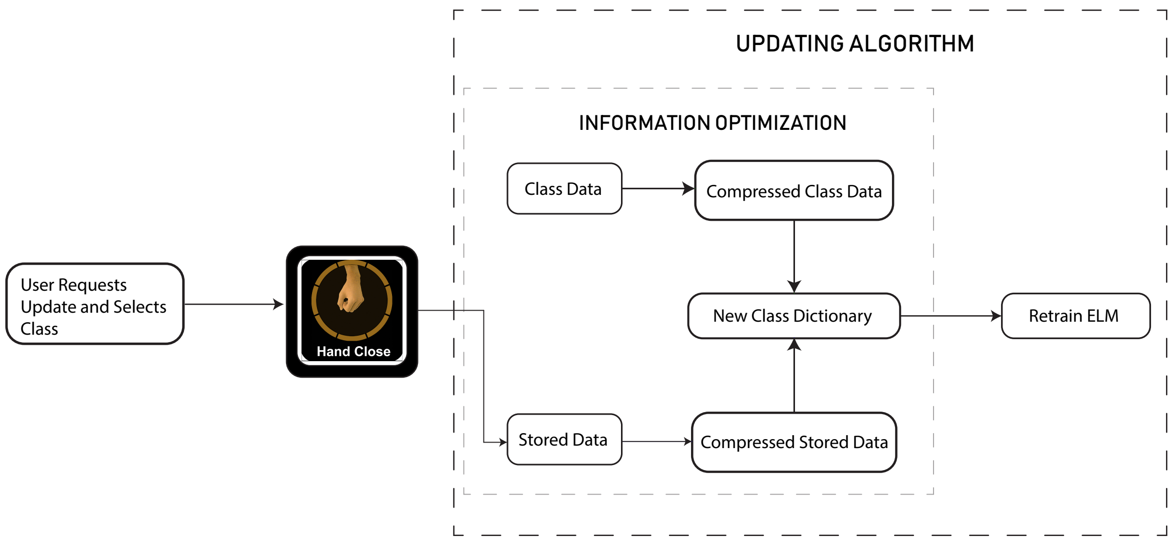

Figure 3.2: An illustration of the general flow of information. The user provides some feedback through a medium and specifies the class. Information that has been stored regarding the prediction of the user's motions is compressed to obtain representative vectors. Then, the class is compressed and these representative vectors are combined to form a new class sub-dictionary. Using this updating dictionary, the ELM Boundaries are then calculated.

3.3.2 Updating Method

After a user selects and relabels the misclassified inputs, K-Means clustering is performed on the buffered data in order to identify K samples that capture the overall structure of the input. Given that a class C has a sub-dictionary size of S_C, the sub-dictionary is reduced by performing $S_C - K$ means yielding a new reduced sub-dictionary size of $S_C - K$ which accomplishes two things: firstly, as a user continues to perform updates, the vector space of the class slowly starts to shift so that sample vectors that have ***not*** been contributing to predictions are filtered and secondly, it also ensures that the size of the sub-dictionary is constant once the K samples from the buffered data are incorporated. This ensures that over-represented data is not incorporated so that the one class does not contribute to uncorrelated classes when the minimized $l1$ is calculated for an input. After incorporating the data, the system is then retrained in order to calibrate the output weights for the ELM. This method of updating focuses on the information perspective that is unique to SRC and then subsequently accounts for the spatial discrimination in feature space for the ELM. This can be observed in Figure 3.3.

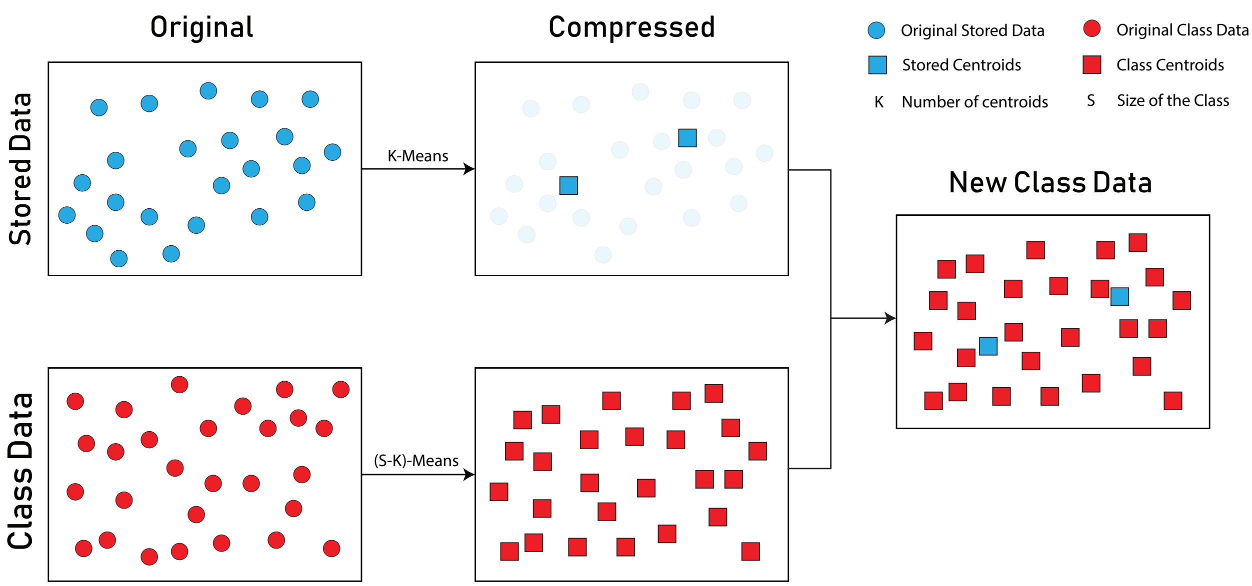

Figure 3.3: Stored Data (top row) depicts data accumulated during an update. Class Data (bottom row) depicts data that exists in the class sub-dictionary of the classifier. The stored data is compressed into K representative samples and the class data is compressed *by* K samples after performing (S-K)-Means. The K representative samples and the compressed dictionary are then combined to form the final class sub-dictionary.

By incorporating representative vectors of entire motions, we not only reduce the need to increase the size of the dictionary, but also reduce the amount of time required to obtain new vectors that capture the user's environment. The presence of these vectors greatly increases the ability to predict any future movements that are of similar conditions and with the addition of old samples, we can retain information from multiple environmental conditions thus, providing more robust predictions.

3.4 Conclusion

The core idea of this chapter is to explain the intuition for the choice of using EASRC as the foundational classifier and K-means for the updating system. To recap, EASRC takes advantage of an ELM to provide fast predictions when it is confident and act as a filter to reduce the number of classes that SRC may have to work with if it is not confident. By using input reconstruction as a basis for classification, SRC is able to provide robust classification and is not nearly as computationally challenged when paired with an ELM. This manner of classification also requires the incorporation of vectors that captures the subject's environment. K-Means is used to capture the K most representative samples from data acquired during an update and incorporate them into the class dictionary. During the update, the class dictionary size is static in order to prevent any one class from becoming asymmetrically more dominant in contribution during input reconstruction. In the next chapter, the updating method is tested alongside the classifier in simplified conditions that reflect the intended user experience.

Chapter 4

Validating Updating Scheme

4.1 Introduction

In order to verify the updating system's ability to accommodate for environmental variation, the experiments focus on classification robustness to changing hand positions. Hand position is a common asymmetric variation that is instrumental to interact with the surrounding world and participate in the activities of daily life for both able-bodied and amputees alike. Hence, the ability to provide consistent or improved classification in different positions can be beneficial for amputees. In order to test the updating method, two experiments were conducted: an offline variant to tune the updating method and an online variant to verify the system's behavior. Here, the offline variant simply collects raw sEMG data from a subject performing a series of classes (grips) at different positions without providing or receiving any feedback from the subject. This data is used to compare the potential for a classifier that does not have an augmented updating scheme (the control) versus a classifier that does (the experimental). These results were then used to optimize for different features and updating parameters in the for the online double-blind experiment, where a subject's subjective feedback is considered.

The offline variant was used as a preliminary study to observe the effects of the updating method with respect to the control. Using the results gathered from the offline variant, certain parameters and experimental design changes were adopted and are reflected in the online variant, as discussed later in this chapter.

4.2 Methods

4.2.1 Experimental Setup

For both, the offline and online experiment, the subject is faced with a similar physical experimental setup. The subject is first equipped with Norotrode DDN-20 disposable surface bi-polar electrodes (Myotronics, Kent, WA) placed equidistant radially around the subject's most muscular region of their forearm; generally 1/5th of the distance from their elbow. A representation of the physical setup is given in Figure 4.1. An additional electrode was placed by the elbow to serve as ground. In order to simulate the weight of a prosthesis that an amputee uses, the able-bodied subject is equipped with a able-bodied cuff that holds a BeBionic hand at the end. A National Instruments (NI) DAQ USB-6009 (National Instruments, Austin, TX) was used to read sEMG data streaming in at 1024 Hz amplified using the 13E200 (Ottobock, Plymouth, MN) amplifiers. This data is processed through the Analog Filters in the 13E200, which has bandwidth from 90-450 Hz. The filtered sEMG data was then feature extracted for TD5 Features and FFT features.

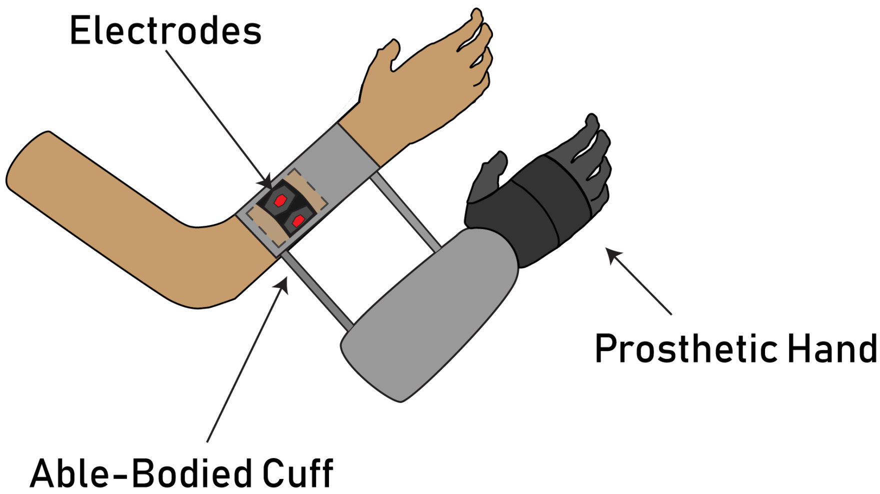

Figure 4.1: Representation of the equipment setup including electrode placement being roughly a fifth of the distance from the elbow, and the relative position of able-bodied cuff on a subject. The able-bodied cuff serves as the load a subject would face when dealing with the prosthesis

The subject is then placed in front of 3x3 board where they are required to train and then test at specified locations as observed in Figure 4.2. Each square represents a 23 cm x 23 cm space that the subject must position their hand towards and perform a series of gestures given by Figure 4.3B.

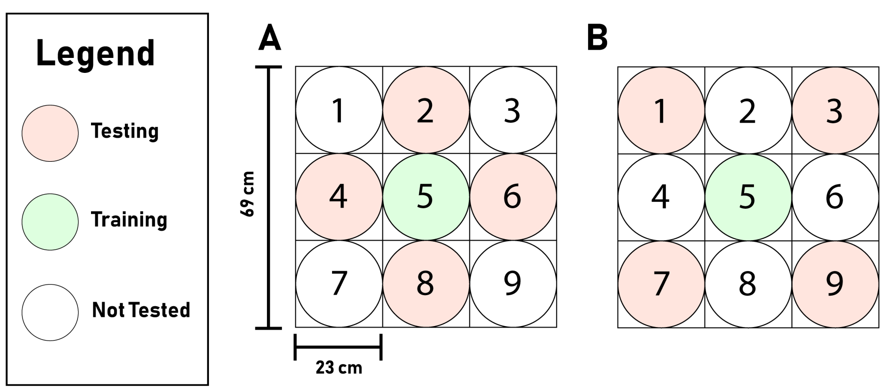

Figure 4.2: The specifications of the 3x3 square board that is used in the position variation experiments. Each block is a 23 cm x 23 cm and is used to guide subjects to consistently hold a particular position over the course of the experiment while they perform different grips. (A) Represents the conditions used for training and testing for the experimental data acquired during offline variant and (B) represents the experimental data acquired during online variant.

4.2.2 Offline Experiment

Two-able bodied subjects participated in the offline experiment. Each subject was positioned such that position 8 acted as a neutral hand position; the elbow was tucked to their side and the arm was bent at a 90 degree angle. This made it such that position 5 behaved like an added 45 degree extension from the neutral position. In the offline variant, subjects were requested to train at position 5 and then test in positions 2, 4, 5, 6, and 8. Position 5 was specifically chosen because it falls in the middle row allowing positions 4 and 6 to be horizontal deviations with a similar gravitational load, whereas positions 2 and 8 present slightly different gravitational loading profiles. This setup is visually represented by Figure 4.2A.

Equipped with 8 electrodes and the able-bodied cuff, the subject began the experiment by first collecting training data. With their hand at position 5 the subject performed each of the 7 classes for 5 seconds with 3 repetitions. The first 2 seconds of the data are ignored and only the last 3 seconds are recorded. The 2 second period acts as a buffer for the subject to recognize and perform the requested class. Ignoring this data also reduces the likelihood of capturing class transition information as a subject changes from one grasp to the next. Thus, the last 3 seconds captured represents the subject performing a static hand grip. This data is then stored and used as the training set for analysis and is only obtained during the very first data acquisition session. All of the 7 classes required by the subject are presented in a randomized order. After acquiring a training set, the subject is then required to obtain four testing sets. For each testing set, the subject is guided through a random permutation of the five testing positions. At each position the subject performs each of the seven classes - in a randomized order - for 5 seconds. The requested class is prompted as an image on a computer screen alongside the position as seen in Figure 4.3A.

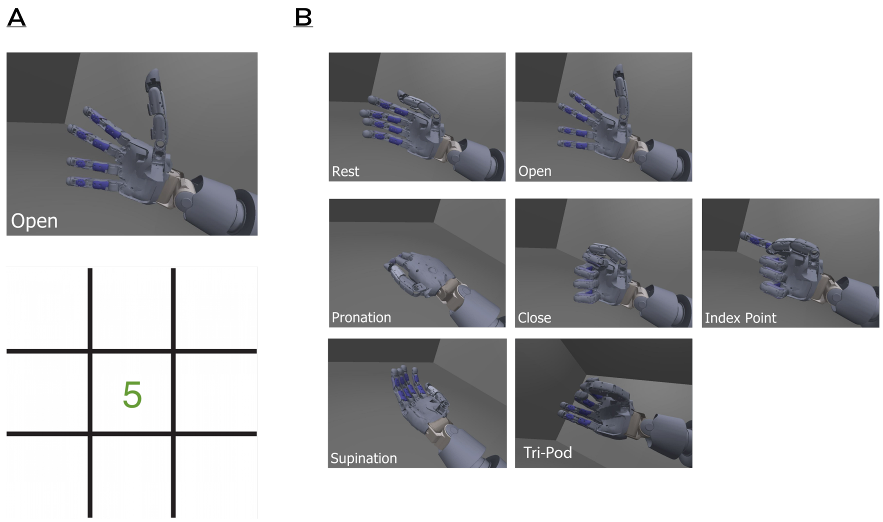

Figure 4.3: (A) Illustrates the images that appear on the computer screen - a pairing of class and position. The subject is given as much time as they need in order to move their hand to a different position and perform a class for the requested duration. (B) Represents the 7 class images the subject sees when performing the online and offline experiments.

We consider a pair of a single position and class as a task, therefore in each trial there are 5 testing positions with 7 classes, or 35 tasks. Three such trials are acquired for each data acquisition session. The subject is given as much time as needed to rest after collecting data at each position in order to reduce the effects of muscle fatigue. For each subject, a total of four data acquisition sessions are obtained over 2 days. One session occurs in the morning and another roughly 6 hours in the afternoon. By the end of the offline experiment, there is 1 training data set and 4 testing data sets for each subject. The raw EMG data was acquired and stored as the subject performed each grip at the specified position, and alongside the raw data, the order of the tasks was also recorded to feed data chronologically during analysis.

Offline analysis involved initializing two EASRC classifiers and seeding them with the same randomized weights and biases for comparison between the control and experimental methods. Both classifiers were initialized using the same training set and were also tested by feeding in the same testing data in the chronological order for one subject at a time. The training data was used to assemble a training dictionary under the constraint of approximately 300 feature vectors. Since the subject is performing 7 classes, this resulted in a dictionary size of 301 feature vectors where each sub-dictionary (dictionary of each class) had 43 feature vectors. This constraint was used to allow the algorithm and updating system to run on small embedded hardware with limited memory. In order to choose the 43 feature vectors for each sub-dictionary, the 3 seconds of data from each of the 3 trials was feature extracted using a window size of 200ms alongside a 25ms window step-size. After feature extraction, the middle 14 feature vectors from two randomly chosen trials and the middle 15 feature vectors from the final third trial were chosen to constitute the class's sub-dictionary. The middle feature vectors were chosen to avoid class transition data and capture sEMG that showed minimal signal degradation. After initializing the EASRC classifiers, the testing data was fed in chronological order - one-task at a time. In the offline analysis, if the classification accuracy of performing a certain class c resulted in the classifier predicting the true class <70% (chosen threshold of acceptable performance) of the time after feeding all of the data for a given task, then an update was called. For the update, the experimental classifier would utilize the entire 3 seconds worth of feature extracted data from a task in order to modify the classifier's training dictionary, whereas the control would observe no change. The flow if this process is captured in Figure 4.4A.

The primary purpose of the offline variant was to verify the efficacy of the updating system and to investigate the differences in performance with respect to the use of certain features (TD5 versus TD5+FFT) at varied dictionary sizes (301 versus 1400 feature vectors).

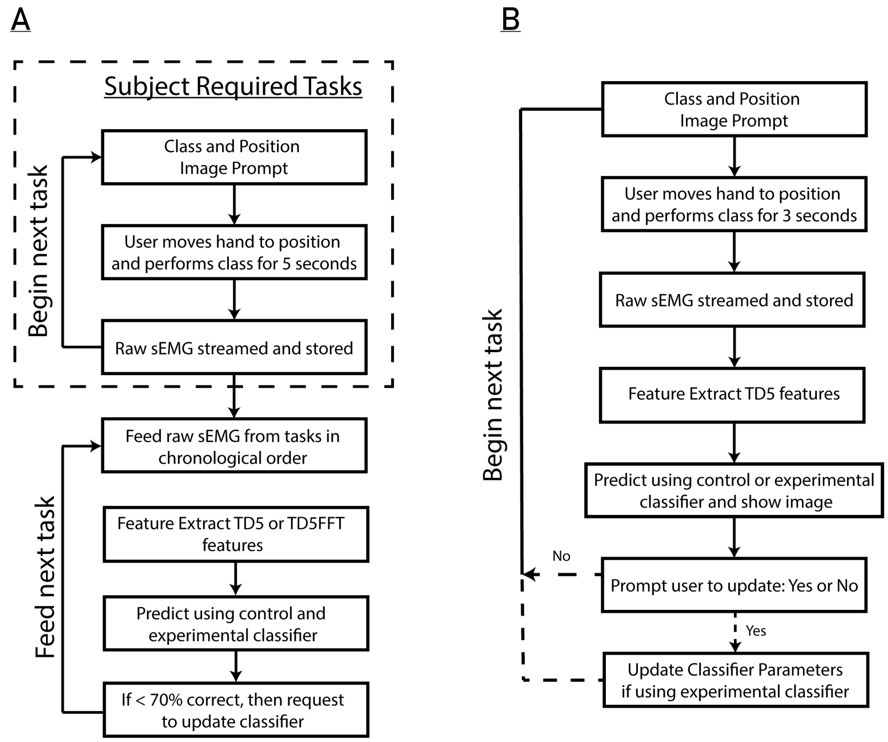

Figure 4.4: (A) Illustrates the workflow of collecting and processing the data in the offline experiment. In this experiment, the subject is only involved in gathering raw sEMG data while the additional data processing occurred offline. (B) The updating in the online variant depends on the feedback obtained by the subject in the form of the yes or no question asked at the end of a task. This online variant helps to reinforce trends that may have been observed in the offline analysis while also seeing how the subjective nature of the feedback affects classifier performance.

4.2.3 Online Experiment

The online experiment extends from the offline variant with some slight changes to experimental procedure. Three different able-bodied subjects participated in the online experiment. Each subject was equipped with 7 electrodes and an able-bodied cuff with a BeBionic hand and was positioned in front of a 3x3 board where position 5 acted as the neutral hand position and the site for training data acquisition. Positions 1, 3, 7, and 9, were chosen to be the testing sites as seen in Figure 4.2B. The reason for change in testing positions was to increase the distance the subject would have to deviate from the training position in order to stress the classifier. Thus, instead of a simple 23cm deviation vertically or horizontally, this now represents a $23\sqrt{2}$ cm deviation diagonally.

Training data acquisition was the same as the offline variant; three sets of raw sEMG data was acquired as the subject performed each task for 5 seconds, where the first 2 seconds are not recorded to account for transitional changes and the last 3 seconds are used to generate the training data. Only TD5 features were used during the online experiment as a means of decreasing feature dimensionality, speeding computation, and simplicity. The final dictionary size was therefore 35 x 301 for 35 feature dimensions (5 TD features from each of the 7 channels) and 301 dictionary samples. Just like in the offline variant, the training data is only acquired during the very first session and is used to initialize two EASRC classifiers: the control and the experimental.

For each classifier, the subject performed 4 testing sessions, where each testing session involved the subject performing 28 tasks in a random order (7 classes and 4 positions) for 3 repetitions. In contrast to the offline variant, the subject does not perform all 7 classes at a single position before moving to a new position; instead, they are given a pairing of position and class in some random order. These 4 testing sessions were performed over 2 days, and on each day, one session was conducted in the morning and another session at least 6 hours later. Thus, for each subject, 1 training data set and 4 testing data sets were acquired.

For each task in a testing session, the subject was provided an image prompt as seen in Figure 4.3A. The subject was required to move their hand to the requested position and perform the requested class for 3 seconds. Simultaneously, the subject was provided visual feedback of the classifier's prediction in the form of images on a computer screen. After completing the task, a yes or no prompt appears with the question "Would you like to update?". The subject must use the visual feedback from the images and the requested class as reference to gauge their performance and choose an option. If the subject elects to update, then the response is recorded and the classifier is updated if the experimental method is being tested; otherwise, the classifier is not updated. Note that instead of requiring the subject to acquire new data in order to perform the update, the sEMG data acquired while the subject was performing the task was used instead. This retrospective nature can potentially capture variations from the subject adjusting their grip in response to the provided visual feedback. In the case of the control, the original training dictionary is used throughout all four sessions since the classifier's training dictionary is not updated in anyway while the experimental classifier changes every time the subject requests for an update. At the start of every new testing session, the experimental classifier utilizes the most recently updated dictionary from the previous session. The only time the dictionaries of the control and experimental classifiers are the same is at the beginning of the first testing session, or an update is not called when testing the experimental method across all four sessions. This online experiment was conducted in a double-blind fashion where the classifier being tested is not known to the subject. The flow of this experiment is given in Figure 4.4B.

4.3 Results

The results section here considers the raw values in the offline and online variant. All updates in the offline and online variants use a K-value of 8. Hypothesis testing was done by calculating two-sample t-test. Throughout this section feature vectors are referred to as samples. When discussing features, TD5 refers to feature vectors with 5 time domain features and TD5+FFT refers to feature vectors with the 5 time domain features and frequency information. Throughout this chapter, a consecutive set of three trials (1-3, 4-6, 7-9, 10-12) correspond to sessions 1, 2, 3, and 4 respectively.

4.3.1 Offline Experiment

Results investigating the classification performance will focus on the constraints of using TD5 features with a dictionary size of 301 samples. The first section will explore the difference in raw performance using the classification system with and without updating. The subsequent section will explore how the classification performance and updating system dynamic changes when using the updating system for four variants of different features (TD5 and TD5+FFT) and dictionary sizes (301 samples and 1400 samples). Exploring these variants provides insights into optimizing for the appropriate number of dictionary samples and feature set with respect to this updating scheme.

4.3.1.1 Classification Accuracy and Updating

Global Accuracies of Classifiers With Varying Dictionary Sizes and Features Over Trials

			Trial													
		Method	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Mean**	**Standard Deviation**
Subject 1	**TD5 only (301)**	**Control**	86.83	92.47	89.91	89.96	87.88	80.10	85.34	81.42	73.99	68.22	73.44	68.05	81.47	8.30
		Experimental	88.15	94.26	92.80	92.12	92.35	89.57	90.58	91.45	90.14	86.58	95.30	91.80	91.26	2.35
	TD5+FFT (301)	**Control**	85.69	88.87	83.48	82.83	82.61	80.32	79.45	75.60	70.88	67.45	70.48	65.91	77.80	7.24
		Experimental	86.98	87.80	84.45	86.61	89.86	88.22	87.93	85.34	87.35	84.17	91.73	88.17	87.39	2.05
	TD5 only (1400)	**Control**	92.40	93.42	92.32	93.19	90.19	85.61	88.45	86.01	76.65	71.35	74.01	68.80	84.37	8.76
		Experimental	93.24	94.11	93.61	91.83	91.28	92.65	90.24	94.48	88.75	85.37	96.22	92.02	91.98	2.76
	TD5+FFT (1400)	**Control**	89.69	91.08	90.46	88.12	88.10	85.94	83.50	81.02	74.78	72.22	76.72	70.56	82.68	7.11
		Experimental	89.57	89.79	90.21	90.21	91.50	88.87	91.73	88.97	86.01	83.70	89.96	87.65	89.01	2.19
Subject 2	**TD5 only (301)**	**Control**	78.01	76.30	64.15	43.68	36.47	32.22	55.93	49.09	47.45	48.42	38.26	41.76	50.98	14.30
		Experimental	78.63	82.53	76.15	69.07	71.90	68.35	73.89	69.66	66.14	68.27	71.88	68.27	72.06	4.70
	TD5+FFT (301)	**Control**	69.86	70.96	58.91	42.98	40.07	34.21	56.42	50.01	42.16	46.76	41.14	41.86	49.61	11.42
		Experimental	71.01	77.74	74.66	62.63	67.25	68.50	64.92	64.75	64.94	67.83	62.61	66.11	67.75	4.46
	TD5 only (1400)	**Control**	78.88	77.39	67.33	36.47	36.17	34.46	47.78	37.71	40.10	36.52	31.85	42.53	47.27	16.41
		Experimental	79.11	81.52	77.29	67.80	74.56	73.22	76.30	72.77	70.36	75.13	71.63	75.58	74.60	3.63
	TD5+FFT (1400)	**Control**	75.90	76.20	64.75	39.80	36.40	31.55	51.33	42.81	42.76	40.60	35.40	39.23	48.06	14.95
		Experimental	77.14	81.66	79.08	62.71	68.50	69.84	70.26	67.33	67.43	70.01	64.89	71.13	70.83	5.46

Table 4.1: Global accuracies for different set of features and dictionary sizes separated by subjects. The standard deviation for the updating method is smaller than the control and the means are consistently higher with the experimental method when compared to the control.

Subject 1 exhibits strong classification averages across trials for both the control and experimental methods. The average accuracy across all of the trials using the control for Subject 1 is 81.5% $\pm$ 8.30% while the experimental yields an average of 91.3% $\pm$ 2.35% (refer to Table 4.1). As a result of the relatively high performance across trials, the experimental method using TD5 features requested for updates 33 times while the control method requested 94 updates across all the tasks (refer to Table 4.2). While subject 2's classification performance was not as strong as subject 1's performance, the experimental classifier outperforms the control by 21.1% with the control averaging 51.0% $\pm$ 14.3% and the experimental averaged 72.1% $\pm$ 4.70%.

Updates of Classifiers With Varying Combination of Dictionary Sizes and Features

		Method	Rest	Close Hand	Open Hand	Pronate	Supinate	Tri Pod	Finger Point	Total
Subject 1	TD5 only (301)	Control	3	11	12	12	20	1	35	94
		Experimental	1	9	4	3	4	2	10	33
	TD5+FFT (301)	Control	0	6	27	3	17	7	59	119
		Experimental	2	8	5	2	5	3	17	42
	TD5 only (1400)	Control	5	8	21	9	14	1	16	74
		Experimental	1	6	5	2	4	2	5	25
	TD5+FFT (1400)	Control	5	8	24	9	18	4	15	83
		Experimental	1	7	4	2	5	3	14	36
Subject 2	TD5 only (301)	Control	49	49	39	14	34	16	37	238
		Experimental	25	40	9	9	8	27	43	161
	TD5+FFT (301)	Control	42	47	44	10	33	33	59	268
		Experimental	26	43	27	13	8	29	58	204
	TD5 only (1400)	Control	48	48	16	8	35	46	52	253
		Experimental	18	33	9	12	12	27	34	145
	TD5+FFT (1400)	Control	49	49	36	5	35	47	55	276
		Experimental	18	46	25	14	12	23	50	188

Table 4.2: The total number of updates called by each class across all four sessions tested with different combinations of features and dictionary sizes. The experimental method requires less updates across both subjects.

Testing with subject 2's data resulted in 161 updates when using the experimental method versus 238 updates when using the control. The p-value for both Subject 1 and Subject 2 when comparing classification accuracies from all tasks between the control and the experimental results are $8.63x10^{-12}$ and $2.84x10^{-20}$ respectively. This difference in performance between the control and the updating method are observed in all variants (as seen in Table 4.3). These results illustrate that each time a user calls for an update, the classifier is able to utilize new samples for improved classification performance.

Classification Accuracy Between Control and Experimental In Variants

	Variant			
	301 TD5	**301 TD5+FFT**	**1400 TD5**	**1400 TD5+FFT**
Subject 1	8.63E-12	2.35E-11	9.66E-09	9.79E-08
Subject 2	2.84E-20	6.38E-18	4.11E-31	4.34E-27

Table 4.3: The p values of the different classification accuracies between the control and experimental for subject 1 and subject 2 under different combinations of dictionary size and features.

4.3.1.2 Effects of Features and Dictionary Size

In this section, classifier performance will be discussed using different features (TD5 versus TD5+FFT) and different dictionary sizes (301 feature vectors versus 1400 feature vectors).

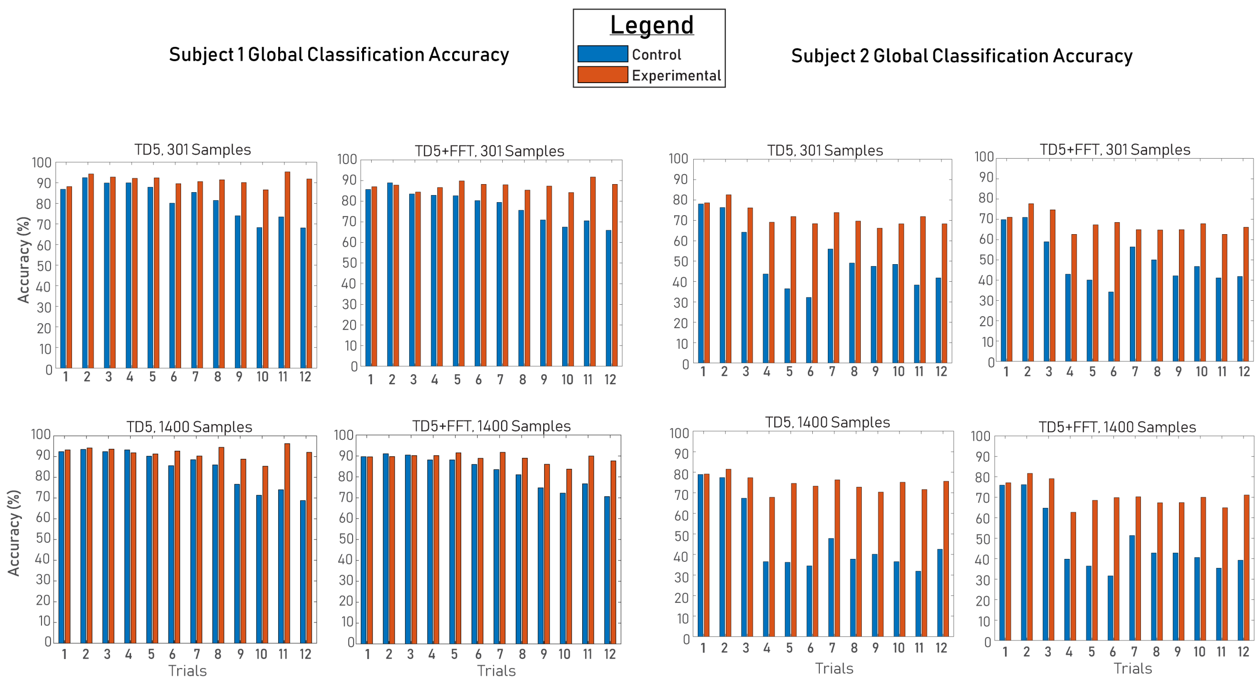

Figure 4.5: A combination of global classification accuracy when using different features (TD5 vs TD5+FFT) and when using different dictionary sizes (301 feature vectors versus 1400 feature vectors). The four graphs represent analysis of offline data using a different combination of these tasks while the four graphs on the right correspond to analysis using subject 2.

The control and experimental classification performance for subject 1 when using TD5 versus TD5+FFT features with a dictionary size of 301 samples yields p-values of 0.03875 and $3.25x10^{-5}$ respectively. Only the experimental method was statistically significant for subject 2 (p = 0.0066) under these constraints. With a dictionary size of 1400 samples when comparing TD5 vs TD5+FFT, the only statistically significant improvements are observed in experimental method with p = 0.00095 and 0.02171 for subjects 1 and 2 respectively. No statistically significant results were obtained for $p < 0.05$ when comparing dictionary sizes of 301 samples to 1400 samples using TD5 features alone, while only the control for subject 1 observed an improvement when having a different dictionary size using TD5+FFT features for dictionary sizes of 301 samples versus 1400 samples with a p value of 0.00225. All statistically significant results are provided in Table 4.4 and classification accuracies across trials can be visualized in Figure 4.5.

Significance of Classification Accuracies Between Different Variants

		Variant			
	Method	**301 TD5 vs TD5+FFT**	**1400 TD5 vs TD5+FFT**	**301 vs 1400 TD5**	**301 vs 1400 TD5+FFT**
Subject 1	**Control**	**0.038753334**	0.268622743	0.089751887	**0.002256606**
	Experimental	**3.25E-05**	**0.000952094**	0.422234589	0.076997147
Subject 2	**Control**	0.591836616	0.757383982	0.169593053	0.518627806
	Experimental	**0.006643434**	**0.021708976**	0.177022343	0.055882099

Table 4.4: Bold entries correspond to significant results where $p < 0.05$. Instead of comparing the control to the experimental, here the control and experimental from one variant are compared to the respective control and experimental behavior in the other variant. This is to find trends that contain some level of significance.

The classification performance difference between controls and experimental methods when using TD5 and TD5+FFT were only significant for $p < 0.05$. The overall performance averaged across all trials for Subject 1 using 301 samples and TD5 features alone results in 3.7% improvement with respect to the control and a 3.9% with the experimental classifier compared to TD5+FFT features. For subject 2, using TD5 features alone results in a 1.4% increase in classification accuracy for the control classifier and 4.3% increase with the experimental classifier compared to TD5+FFT features. This trend holds yet again when comparing the dictionary sizes at 1400; subject 1 observes an 1.7% improvement with respect to the control and a 3% improvement with respect to the experimental while subject 2 observes a 1.1% *decrease* in performance with control, but a 3.9% increase with the experimental using TD5 features versus TD5+FFT features. The classifier configurations therefore suggest that the increased resolution of TD5+FFT features may result in a slight decrease in classification performance. Additionally, the subject would have to spend a greater amount of time requesting for updates if they used a classifier with TD5+FFT features and dictionary size.

The primary difference in performance is observed in the number of updates requested when using different combinations of features and dictionary sizes. From Figure 4.6 and Table 4.2, the control requires 25 more updates when using TD5+FFT features versus TD5 features with a dictionary size of 301 samples and 9 more updates with 1400 samples for subject 1. This behavior extends when using the experimental method, where subject 1 requires 9 updates more using TD5+FFT with 301 samples, and 11 more updates when using a dictionary size of 1400 samples. Subject 2's data shows the control requires 30 more updates using TD5+FFT Features with 301 samples and 23 more updates when using TD5+FFT with 1400 samples. The experimental method suggests 43 more updates were required with a dictionary size of 301 samples and 1400 samples. These differences suggest that using TD5+FFT features does not provide significantly different classification performance, but would result in a greater number of updates to be called by a user. Thus, for an amputee the control performance would increase relative to the control, but the amount of time spent requesting for updates would also increase.

When comparing the performance in dictionary sizes, subject 1's offline data would suggest that the control would require 20 less updates when using TD5 features with 1400 samples, and 36 less updates when using TD5+FFT features with 1400 samples. Similarly, the 8 less updates are required with TD5 features, and 6 less with TD5+FFT features at 1400 samples when using the experimental method. Subject 2's data suggests that the control with TD5 features the difference in dictionary size results in 15 more updates with the larger 1400 samples; while TD5+FFT requires 8 more updates when using 1400 samples for the control. Using the experimental method, the larger dictionary size requires 16 less updates with a dictionary size of alone with 1400 samples for TD5 and TD5+FFT features. These differences would suggest that increasing the dictionary size can reduce the number of updates called. This implies that a prosthesis user spends less amount of time updating if the classifier was running with a larger dictionary size.

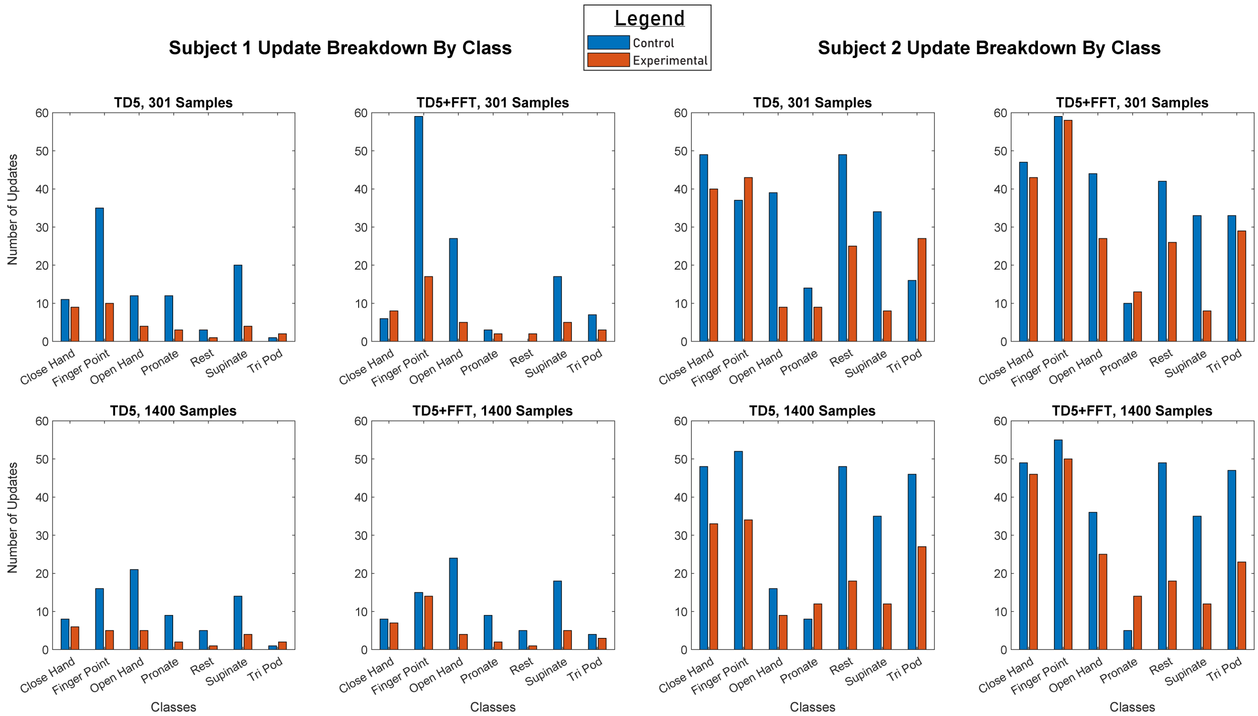

Figure 4.6: Offline analysis suggesting the number of updates called per class broken down by the four different combinations of the features (TD5 vs TD5+FFT) and dictionary sizes (301 vs 1400.)

4.3.2 Online Experiment

In the online variant, the subject is given visual feedback in the form of class images while attempting to complete a task. From this visual feedback, the subject must make the decision on updating their classifier. The control method does not change the sub-dictionary while the experimental method incorporates new elements after each update. This section strictly focuses on the results obtained when using a dictionary size of 301 samples with TD5 features across 3 able-bodied subjects. Note that subject 1 and 2 presented in the online experiment are different people from subjects 1 and 2 from the offline experiment. In this online variant, subject 1 has moderate experience with myoelectric control, subject 2 has had a significant amount of time with myoelectric control, and subject 3 is entirely new to myoelectric control.

Global Accuracy In Online Experiment Across Subjects

		Trial													
	Method	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Mean**	**Standard Deviation**
Subject 1	**Control**	71.21	67.98	70.42	65.69	65.41	51.98	69.47	62.07	53.88	14.93	14.63	14.49	51.85	22.21
	Experimental	80.37	88.87	92.25	85.82	95.65	91.57	87.27	93.11	90.39	72.84	94.12	91.62	88.66	6.20
Subject 2	**Control**	64.67	67.73	48.22	54.41	63.43	68.73	67.71	60.67	69.43	56.08	49.96	47.96	59.92	7.90
	Experimental	86.38	88.99	87.55	75.09	94.41	96.37	93.08	93.03	93.43	94.99	96.99	96.91	91.44	5.98
Subject 3	**Control**	69.02	59.64	72.46	56.43	69.38	76.65	57.30	57.87	36.17	73.78	67.63	70.44	63.90	10.68
	Experimental	77.19	89.12	86.70	74.71	83.11	91.81	83.38	90.16	87.87	88.81	92.87	93.05	86.56	5.67

Table 4.5: Global accuracies of the control and experimental method across all 12 trials for each subject. The mean and standard deviation are provided and follow a similar trend that was observed in the offline analysis. Here we observe that the experimental method has the smallest standard deviation for all three subjects.

In the online experiment, subject 1 had an average classification performance of 51.85% $\pm$ 22.21% for the control and 88.66% $\pm$ 6.2% for the experimental. Subject 2 has an average classification performance of 59.9% $\pm$ 7.9% and 91.4 % $\pm$ 5.98% for the control and the experimental method respectively. Subject 3 has an average classification of 63.9% $\pm$ 10.7% and 86.6% $\pm$ 5.67% for the control and experimental respectively. The differences in experimental and control methods are statistically significant ($p < 0.0001$) with values of $1.72x10^{-35}$, $2.35x10^{-29}$, and $8.06x10^{-16}$ for subjects 1, 2, and 3 respectively. All averages are presented on a trial by trial basis in Figure 4.7 and in Table 4.5.

Total Number of Updates Called During Online Experiments

	Method	Rest	Close Hand	Open Hand	Pronate	Supinate	Tri Pod	Finger Point	Total
Subject 1	Control	37	14	43	43	31	38	45	251
	Experimental	3	2	31	11	20	34	37	138
Subject 2	Control	21	8	28	48	29	43	24	201
	Experimental	10	5	18	16	20	20	18	107
Subject 3	Control	10	47	44	26	7	38	46	218
	Experimental	3	47	36	19	18	39	42	204

Table 4.6: A breakdown of the number of updates called by the control and experimental method by the subjects involved in the experiment. The updates called by each method is separated by class.

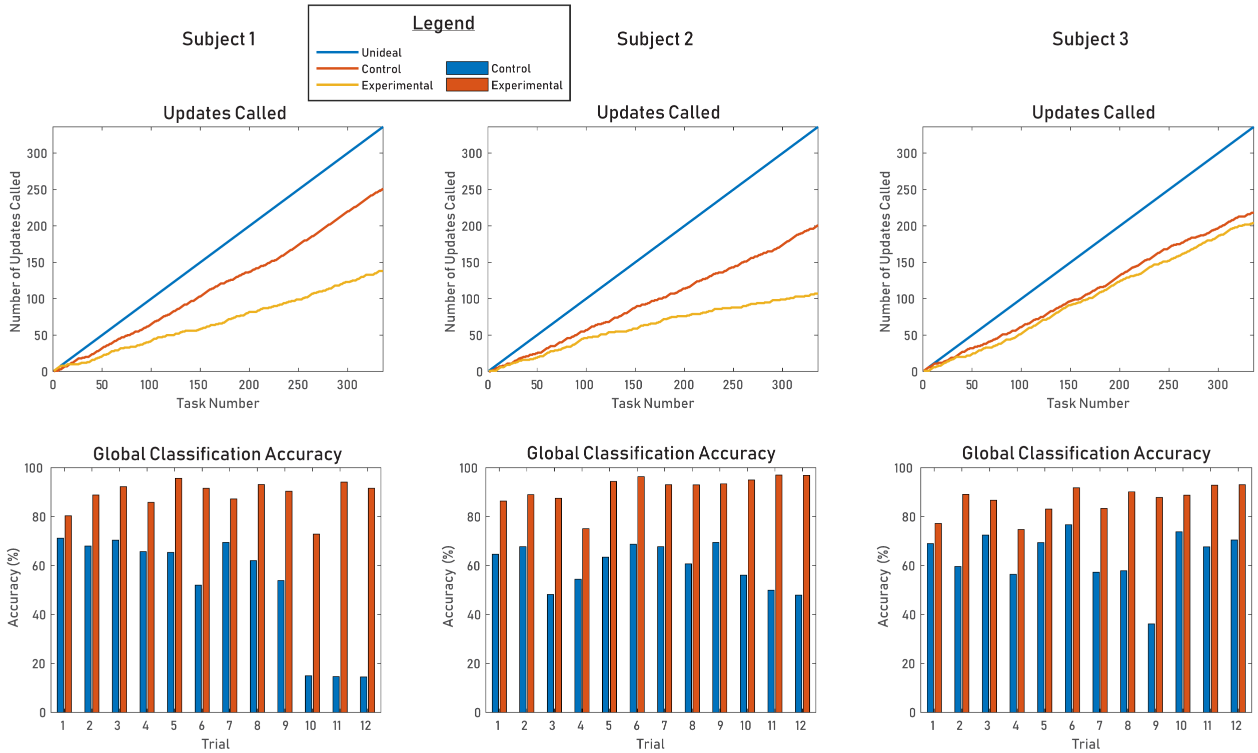

Figure 4.7: Each column represents a subject's data. The top row shows the updating profile for each method during the experiment where the yellow line represents the experimental method and the orange the control over all tasks. The bottom row captures the global classification accuracies over trials for both the control (blue) and experimental (red) methods.

Throughout the experiment, the control method requests more updates than the experimental across all 3 able-bodied subjects. Subject 1 requests for a total of 251 updates using the control and only 138 with the experimental method. Subject 2 requests for a total of 201 updates using the control versus 107 updates using the experimental method. In contrast to the difference in the number of updates called by the control and experimental methods for subjects 1 and 2, subject 3 requests for 218 updates using the control and 204 when using the experimental. The number of updates and the average classification accuracy called by each class is given in Figure 4.8. The distribution of accuracies everytime an update is called for a class across all subjects is represents in Figure 4.9.

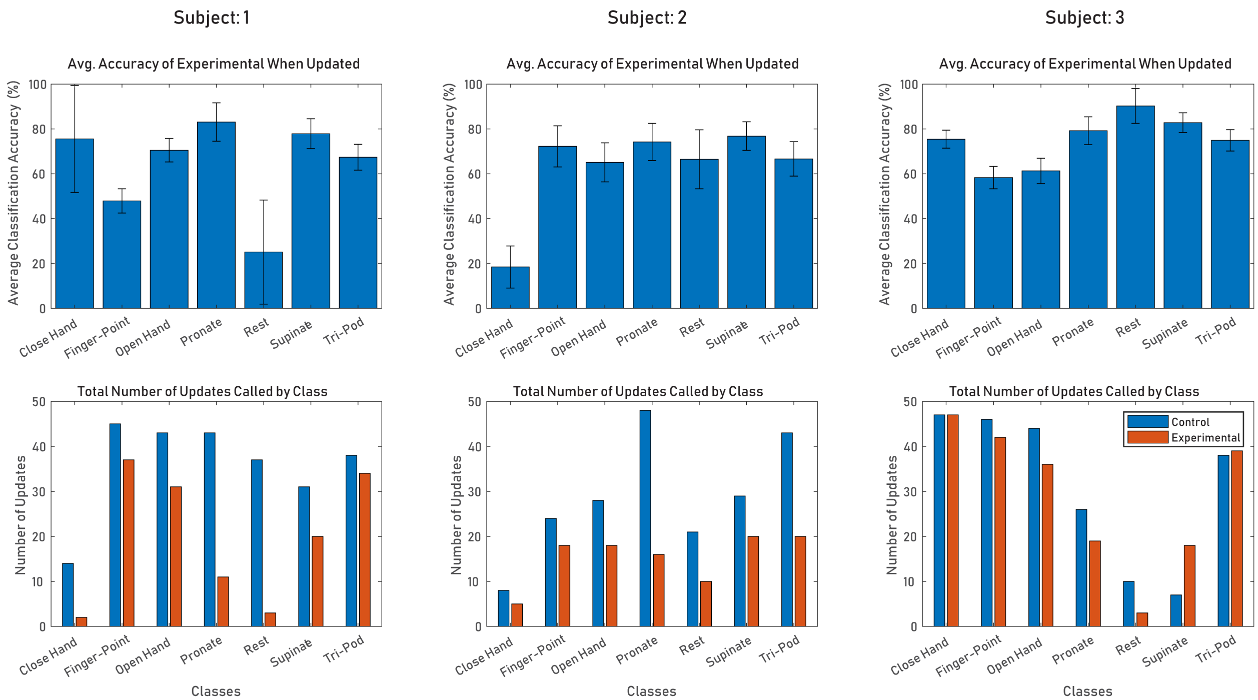

Figure 4.8: Each column in this figure corresponds to data from a single subject. The top row reflects the average classification accuracy over all updates for each class. The error bars are the standard error of the mean. The bottom row is the distribution of updates by each class over all sessions.

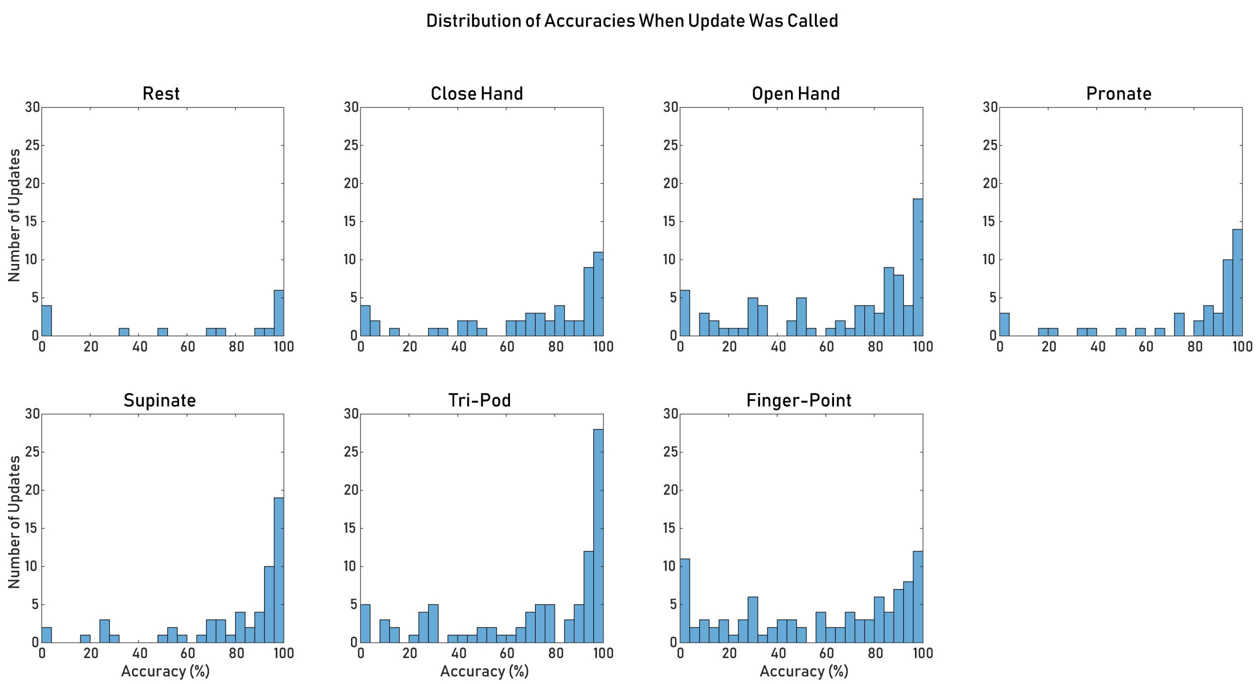

Figure 4.9: This figure represents the distribution of updates called with respect to accuracy by class. The data presented in each class is a combination of three able-bodied subjects. This is used to see at if there is some threshold accuracy where individuals request for updates.

4.4 Discussion

4.4.1 Offline Experiment

4.4.1.1 Classification Accuracy and Updating

There are several trends that are immediately observable when comparing the classifier performance of the control and the updating method from Figure 4.5. We observe an overall decrease in the classification accuracy in the control, but a relatively stagnant performance from the experimental. This behavior observed by the control can be explained by the changing signal characteristics from day-to-day activities. Since the control classifier is never updated, these ever changing signal characteristics lead to the inevitable decrease in performance capabilities of the system. The experimental method however, is an adapting system that incorporates new elements reflective of the subject's environment helping in predictions. This led to a relatively stable accuracy across sessions, and even improved accuracy at the end of the last session. The ability to maintain the improved classification performance led to a decrease in the variance, exhibited by the lower standard deviation, across all subjects when using the experimental model (refer to Table 4.1). Improved performance consequently results in less updates being required as the 70% threshold is not reached as often. This improved performance is what allows the experimental method to make 61 less updates for Subject 1 and 77 less updates for Subject 2.

It is unlikely that the fall off in performance observed by the data can be attributed to a single factor alone; rather, a multitude of factors like the physical activities the subject engaged in leading up to subsequent sessions could have attributed to the signal change in addition to nonstatinarities. One downfall is if the subject is unable to recover greater than 70% accuracy when performing a specific class, then the system will request for an update because the threshold was not reached. This results in asymmetrically more updates when performing certain classes as given by the updating profile of close hand, finger-point, and tri-pod for subject 2 in Figure 4.6. Ideally then, one would except by using the updating system, the classification accuracy of the experimental method would exceed this threshold for all classes as they are updated; however, this phenomenon is not observed even though the classes aforementioned show improvement relative to control in number of updates called. In contrast, classes like close hand, pronate, and supinate which manage to maintain an average classification accuracy greater than 70% over the course of 4 sessions illustrate less required updates for subject 2.

The experimental method for subject 1 requires less updates suggesting each update had greater efficacy. This contrast in updating profiles between both subjects can be explained by the subjects' myoelectric control experience. Experienced subjects perform classes in a way that maximize class distinction; thereby limiting extensive overlap that may exist in feature space. Thus, any update to a class may enhance the subject's ability to perform said class, especially if the update leads to greater class separation or provides greater definition to the subject's immediate environment. For an inexperienced myoelectric user like subject 2, the inability to distinguish between the classes would suggest extensive overlap of certain classes in vector sub-space making it difficult for the classifier to generate an accurate prediction. We observe in Figure 4.5 and in Figure 4.6 that the experimental classifier does improve relative to the control suggesting performance increase and greater distinction in struggling classes. Yet, certain combinations of features and dictionary size would suggest that the number of updates would be greatly increased. The inability to cross the threshold as indicated by the average classifications and the number of updates by the respective classes would suggest the vector spaces are not distinct enough even after updating to improve performance - thus suggesting dependence on user experience and habits. From the initial offline analysis, it is clear that there is a statistically significant improvement ($p < 0.0001$) in performance when the updating system is employed versus one where there is no updating system available in the presence of the limb-position effect.

4.4.1.2 Effects of Features and Dictionary Size

When comparing the different features at some dictionary size, the experimental classifiers indicate statistical significance $p < 0.05$. Consider the breakdown of updates called by classes in a comparison of TD5 with 301 samples versus TD5 with 1400 samples or TD5+FFT with 301 samples versus TD5+FFT with 1400 samples as given in Figure 4.6. Here the TD5+FFT feature set requires more updates in total for both subjects when comparing control and experimental methods across the different dictionary sizes. In order to understand why, consider the dimensionality of the system in both cases. A dictionary size of 301 samples suggest that each class (given equal representation as they are in these experiments) has 43 feature vectors. Using only TD5 features across the 8 channels of raw sEMG that is gathered, each feature vector has a dimensionality of 40. Thus, the sub-dictionary for each class is over defined and as a result the number of samples in each sub-dictionary provide greater definition to the vector space occupied by each class. By adding additional features in the form of FFT to the feature vector, the dimensionality of each sample is increased to 40 time domain features and 512 frequency features by using an FFT length of 128 across the all 8 channels. Therefore, the total dimensionality of a TD5+FFT feature vector is 552 dimensions. Now, the vector space for each sub-dictionary for either scenario is underdefined and even under the assumption that all of the vectors within a vector space were all linearly independent, the span of the vector-subspace does not define all 552 dimensions. The constant class sub-dictionary means that each update will not add any additional definition to a class and as a result, may not have the same efficacy. This sparser data set affects the classification performance and means more updates are required when using TD5+FFT features versus TD5 features alone as seen in Figure 4.6. Thus, in order to assist myoelectric control performance, it is more beneficial to utilize TD5 features because they can be spanned by the vector sub-space of each class. The lack of sparsity assists the updating system when new vectors are introduced.

When comparing the classification performance with different dictionary sizes, there are no statistically significant results when using TD5 features or TD5+FFT features with the exception to the control for Subject 1 ($p < 0.05$). In contrast, the number of updates required when using a larger dictionary size of 1400 samples (compared to 301 samples) is less across all subjects and classifiers, with the exception of the control for subject 2 (refer to Table 4.2). A decrease in the number of updates is also observed for particularly troublesome classes. Consider finger-point for subject 1. Using the dictionary with 301 samples results in Finger Point requesting updates 10 times for the experimental and 35 times for the control - effectively an entire *session* worth of updates. By increasing the dictionary size to 1400 samples, the number of requested updates for finger point drops to 5 updates made by the experimental method, and 16 by the control. This decrease in number of updates implies that the ability to predict finger point more than 70% of the time is greatly increased. This can be explained by the addition of 153 samples in each sub-dictionary (200 samples per class sub-dictionary). This increased number of samples allows the updating system to retain information without necessarily losing historical data in the process of maintaining a constant class dictionary size. This allows the class dictionary to be more robust to the previously encountered environmental changes and would explain the decrease in the number of updates requested by each class. Note that subject 1 is an experienced myoelectric control user who has a distinct signals when performing each class allowing this subject to request for fewer updates and maintain a higher classification accuracy (refer to Figure 4.6 and Figure 4.5).

Looking at Subject 2's classification performance when using TD5 with 301 samples and comparing performance when using TD5 with 1400 samples, we observe no statistical significance as indicated in Table 4.4. However, Subject 2's control exhibits a need for *more* updates with 1400 samples than it does with 301 samples. This is in contrast with subject 2's experimental method which request for *less* updates. The observed behavior can be explained by subject 2's inability to perform distinct enough hand grips to provide distinction in feature space for the vector sub-space of each class. It is likely that there exists large overlap in the the vector-sub space of each class. With a dictionary size of 301 samples, the number of feature vectors overlapping in the 40 dimensional space is likely to be far less than the number of feature vectors overlapping with a dictionary size of 1400 samples. Therefore, an already overlapped sub-space is likely resulting in further overlap leading to greater confusion and thus more updates. The issue arises when the vector sub-space spanned by one class is larger than the class that it overlaps with. The class with the larger sub-space is likely predicted with a greater degree of accuracy than the class that it encapsulates. The most striking trend is that the experimental method for subject 2 shows a trend in the requested number of updates that is consistent with subject 1 who has greater separation in their patterns; the number of updates requested by subject 2 with a dictionary size of 1400 samples is less than the experimental method with a dictionary size of 301 samples. This suggests that the updating method is able to not only adapt to the changing environment, but in doing so, is able to provide some level of distinction between classes.

4.4.2 Online Experiment

From Figure 4.7, the performance observed in the offline analysis is also seen in the online variant: the experimental method performs better than the control method. The primary difference occurs in the number of updates called by each subject. Subject 2's ability to discriminate between classes is seen in a decreased number of updates after subsequent sessions. This would suggest that the elements of the experimental dictionary are becoming more robust to positional changes. This updating profile for experimental versus control is similar to the profile from Subject 1 in the offline analysis - another subject experienced in myoelectric control. While Subject 1 does not show the same experimental profile from cumulative updates as Subject 2 (refer to Figure 4.7), there still exists a large difference in the number of updates called between the control and experimental. In contrast, Subject 3 calls for updates using the experimental method almost as often as the control (refer to Table 4.6). This variance in updating dynamics from subject to subject can be attributed to each subject's experience with myoelectric control.

Looking at subject 3's updating profile, there were a greater number of updates called for close hand, finger-point, and tri-pod (refer to Figure 4.8). These findings are consistent with the visual feedback during the experiment where subject 3 complained about the confusion between these classes. These results are expected considering those 3 classes have similar degrees of freedom in hand movement likely resulting in overlap in feature space. For subject 3, if one of those classes is performing well, the others were performing worse. It is worth noting that if a class A is being confused for Class B when trying to perform A, there is no necessity to suggest that class B will be confused with A, when trying to perform class B. This is possible if the vector subspace for class A is effectively encapsulated by the vector subspace for class B.

Subject 1's performance was a distinct from the other subjects' performance based on the control. Notably, Subject 1 experienced a sharp decline in the control performance by the fourth session while subject 2 and subject 3 were able to maintain or saw minimal decrease in performance over trials. During subject 1's fourth session, when the subject was performing any class the classifier would only predict tri-pod unless disrupted by very strong muscular contractions. For subject 1, the difference in classification performance suggests that by the fourth session the control dictionary does not characterize the sEMG signals reflected by the subject's environment. This behavior is likely a combination of the subject's prior physical activities and potentially due to multiple asymmetric conditions.

Unlike subject 1, subjects 2 and 3 have a more stable control classification; control either maintains or shows minimal decline in performance across trials suggesting the sEMG data was still captured by the control, yet not as well characterized as the experimental method (refer to Figure 4.7).

In order to get a better understanding of a subject's behavior for updating, we can look at the distribution of accuracies when an update was called. In Figure 4.9, we see that the original 70% threshold used in the offline analysis does not hold in the online scenario. Contrary to intuition which would suggest subjects are more likely to request for an update when the performance is poor, the distributions of updates called by the experimental method against accuracy shows most of the updates were requested even at relatively high performance. This is due to two main factors: the average classification accuracy and latency in visual feedback.

Firstly, the experimental method shows an average classification accuracy of 80% or greater in most trials for all subjects. This means that during the course of the experiment, the subject spends a greater amount of time performing certain hand grips with relatively high performance throughout the experiment. As a result of the subject spending more time performing each class with greater classification accuracy, every time a subject requests for an update, the average accuracy of the requested update is going to be skewed towards the higher percentiles. It is also likely that the subject becomes accustomed to their improved performance while trying to perform a class, thus resulting in an increased threshold. Secondly, in order for a subject to gauge their performance, the only visual feedback provided is the image corresponding to the prediction made by the classifier. No additional information in the form of duration elapsed during a task nor their accuracy as a percent is presented. During the task, a classification is occurring every 25ms and each predicted class's image is shown to the user. However, the time it takes to show the image to the subject may take longer than it takes to classify. As a result, while the new image is being presented, classifications are occurring in the background. Since no information is presented to the user as to where they are temporally within a single task, they will have no context as to the number of classifications that have happened relative to how many are left. Consider a transition like close hand to finger point to close hand. If a user was performing close hand for 1 second then misclassified to finger point even once, and then went back to close hand; in the amount of time it takes to show the finger point image and then transition back to close hand, multiple classifications may have occurred. A subject may therefore think the duration of finger-point is longer than it seems and assume their performance was worse than it actually is - especially when the true class is predicted between a series of misclassifications. If this occurs fast enough, the visualization may affect how they view their performance. Together, these two main factors are likely responsible for the distribution of updates spread across accuracies as seen in Figure 4.9. Based on the following, there is no immediate indication of an acceptable threshold performance.

From the online analysis, we observe that the experimental classifier utilizing the information centric updating system is capable of exhibiting improvements for all subjects including subjects inexperienced with myoelectric control. However, the amount of time spent requesting an update is different across all subjects. Experienced users capable of making more distinct EMG signals spend less time performing updates than subjects who are not as capable of distinguishing their signals. Furthermore, there is no indication of an accuracy threshold at which subjects prefer to call for updates as distribution of updates with respect to accuracy is skewed to the right.

4.5 Conclusion

From these results, one major theme that appears multiple times is that the updating the classifier with an information centric focus improves over the control significantly ($p <$

0.0001). Implementing this systems into prosthesis would favor lower dimensional features when using less samples for each class sub-dictionary. If using high dimensional features, the increased resolution needs to be balanced with more samples to span the feature space or there is a detrimental effect to the updating dynamics. Additionally, the increase in dictionary size does not affect the classification performance, rather, it may decrease the number of updates that a subject may call (relative to the control) because more data is available for each class in order to make a classification decision. While the greater number of samples would benefit a user in the decreasing the time spent updating the system, they may observe an increase in the amount of time spent generating a prediction. Therefore, it is important that an optimum is reached between the computation time and the number of feature vectors in a dictionary. Based on the online results, the total number of updates called by a subject is highly dependent on the myoelectric control experience of the subject. Therefore, an inexperienced user may not necessarily tolerate the amount of time spent updating for the improvement in accuracy. This can be addressed by training the user to activate as many different muscles as they can consistently when performing classes to provide greater distinction in feature space. Since only 3 seconds worth of data is used for each update, the amount of time a subject would need to spend retraining is significantly curtailed. The performance improvement also suggests that there is no need to perform an entire training protocol. Instead, performing updates individually to optimize troublesome classes is sufficient to improve classification performance globally. The use of EASRC also suggests that the updating does not have to occur constantly and with large amounts of data.

Chapter 5

Future Works and Conclusion

5.1 Future Works

Given the promising results with able-bodied subjects in both, the offline and online experiments, there are several questions that are raised and should be investigated either individually or pursued in parallel.

5.1.1 Amputee Data Testing

While the online data seems to corroborate the offline analysis, it should be emphasized that all the participants were able-bodied subjects. Amputees all have rather unique physiology due to different surgical procedures and varying degrees of amputation which consequently affects their performance compared to able-bodied participants. Therefore, the most immediate step would be to extend the subject pool to include amputees to explore how their behavior and ability to control their residual musculature can affect the updating system's ability to cope with variances. It is possible that certain patterns exhibited by able-bodied participants will not observed when tested with amputees. Pursuing any future experiments with amputees will still begin with the same hypothesis: the updating system will improve the global classification performance for any subject relative to the control. However, as discovered in the offline and online variants, the number of updates required for each subject will be different based on their prior myoelectric control experience.

5.1.2 Task Completion

In chapter 1, it was suggested that classification performance does not necessarily translate to improved ability to complete tasks requested by TAC or other task completion metrics. While this is true, it cannot be dismissed that the inherent improvement in classification, especially to classes that may be confused for certain subjects, does not translate to more reliable activation of the involved classes. Using the information gained from performing the offline and online testing, it would be appropriate to have subjects perform classes in the context of completing specific tasks. The hypothesis in this scenario is that due to non-stationarities in sEMG signals, the classification performance will degrade over time. Without some method to augment the system, the decrease in classification performance will therefore affect a users ability to complete tasks. Thus, an updating system will be able to rectify for this decrease in classification performance and should assist a user with their task completion. This experiment should be completed with able-bodied and amputee subjects in order to verify whether task completion is improved as a result of improved classification. We expect improved classification and task completion from the offline and online analysis results presented before.

5.1.3 Updating For Task Completion Assessments

In the online experiment, each task required the subject to perform one static class to gather 3 seconds of sEMG data at different positions. After the subject was done completing the task, the subject was allowed to update. In this experiment, the data acquired as the user was performing the task was used to perform an update; the updating data was retroactively chosen instead of collecting new data. In a 1-DOF task completion test, a user will have to perform multiple classes in order to complete a task. For example, a common 1-DOF TAC test shows a virtual hand with some translucent cursor. The subject's goal is to perform a particular class in order to move the translucent cursor to an opaque target cursor and hold the position for 2 seconds in order to complete the trial. The subject is given 10 seconds to complete each task and if the subject is unable to complete the trial within 10 seconds the trial is deemed unsuccessful. During the task, the subject may overshoot the target cursor and will need to adjust by performing the antagonist class. For example, if a subject is informed to perform close hand in order to complete a trial but over-extends past the final target cursor, then the subject is required to perform open hand in order to match the target position in order. As a result, it is difficult to consider classification accuracy alone as a means of verifying performance as knowing which class the user is intending to perform is not easily known. Therefore, in task completion tests, Fitt's parameters are utilized to verify the subject's performance. The monitored parameters include metrics of measuring overshoot, path efficiency, and completion time. Using the following metrics, the updating system can be verified by requesting users to complete update based on their performance at the end of a task.

5.1.4 Autosegmentation and Retrospective Updating for Task Completion Assessment

Note that because of the difficulty in knowing when a user starts and stops a motion, it is difficult to look at the acquired data and provide labels simply by using the raw sEMG. Instead, the data needs to be actively monitored during a trial in order to segment the data in real-time by detecting changes in the gathered sEMG data in order to provide suggestions for retrospective updating. There are three major types of movements that would need to be properly segmented: rest to active class to rest (long contractions), rest to active class to rest (burst contractions), and active class to active class transitions. By detecting these transitions, the duration of the segment and the region of sEMG data associated to a hand grip can be presented to the subject so that the segmented region and all of the elements can be relabeled true class label and then incorporate into the classifier. While this may seem an appropriate way to accomplish the tasks, it is worth noting that the one downside to this process is that the metrics of path efficiency, overshoot, and completion time are highly dependent on how a user performs during the task. If the ability to perform the task is changed mid-experiment, attributing the effects of the updating system to performance becomes a little difficult to quantify. The main advantage to the use of autosegmentation in conjunction to retrospective updating is that new user data can be incorporated using continuous spatial information instead of using data limited to discrete locations as was suggested in the previous sections. By performing tasks in the physical space instead of a virtual space, continuous spatial data can be collected unlike virtual tests like the computerized TAC which does not respond to the physical movement of the arm. Thus, continuous spatial information can be sampled when updating the classifier. Nevertheless, it is very important to note that generating an effective autosegmentation technique is not a trivial task and needs to be done well, for a single poor segmentation used for an update can lead to poor performance and a feed-forward scenario breaking down the classifier entirely. Therefore, potential fail-safe mechanisms to augment an autosegmentation technique are required.

5.2 Conclusion

While there are many nonstationarities that affect the performance of sEMG data, robust classification performance to some, like limb-position, are imperative for users to engage in activities of daily living. Often times, these nonstationarities in addition to the inter- and intra-day variations in sEMG characteristics are responsible for the inevitable breakdown in a classifier's performance. The classifier's breakdown is often compensated for by having a subject perform entire training sessions to initialize a new classifier more representative of the subject's immediate environment. This often leads to a large amount of time being spent on training the pattern recognition system due to frequent retraining. In order to improve classifier behavior, different control methods and updating systems are employed to prevent performance degradation. Control methods try to identify a user's intent and utilize a criterion in order to determine if a user intended to perform a particular hand grip. Satisfying the appropriate criterion will result in the new predicted hand grip; otherwise, the system defaults to rest or the previous motion class. These control strategies do not change the inherent ability of the classifier in generating predictions because they do not modify any of the classifier's parameters. In order to change the predictions themselves, an updating system needs to be employed. Updating algorithms need to be tuned for specific classifiers, although they share one fundamental property: to modify the classifier's core optimized variables. In the updating systems explored in Chapter 2, the LDA classifiers were always updating by modifying the means and pooled covariance - both variables that are required for generating predictions. From the previous works, two main approaches can be used when updating a classifier: by using small calibration data sets or by continuously updating the classifier. While the small calibration sets were often used to compensate for inter-day variations, the contribution of the calibration set's means and pooled covariance were weighted more than the classifier's means and covariance at the time of updating. This places an inherent emphasis on the calibration data when modifying the parameters while also suggesting the need for historical data in generating robust predictions. The downside is that these updating systems were not experimented in the presence of nonstationarities.

The principles from the previous updating methods were used with EASRC - a hybrid classifier that takes advantage of the speed of an ELM and the robust prediction of SRC. The ELM is used to generate a prediction when it is confident, while behaving as a filter to reduce the number of classes SRC needs to work with when it is not confident. The dictionary reduction is required as SRC's computation time increases proportionally to dictionary size which must be actively maintained in order to provide a prediction. This interplay between the two systems allows EASRC to be inherently more robust in classification while still behaving within the real-time constraints required in the EMG problem space. In order to generate a prediction, SRC places an emphasis on input (being a feature vector) reconstruction based on the samples that constitute the classifier's dictionary. In doing so, the system returns the class that contributes the most to the input reconstruction implying a dependence on information content. Therefore, when devising an updating system, the focus should be on optimizing for this information criterion by incorporating representative samples into the actively maintained dictionary. This way, future instances reflecting the same environmental properties faced by the subject are more robustly classified. To obtain representative samples, a subject would perform a class they would like to correct for 3 seconds. This data would then be sampled by using K-Means with a K value of 8 to capture some structure of the input and then incorporate these 8 new feature vectors after compressing the class sub-dictionary by 8 samples. Using this kind of system avoids the need for a subject to completely retrain their classifier and also provides the subject control over when and what they wish to update.

Two experiments were conducted to explore the effects of the updating system on classification performance to variations in limb-position. In order to understand the effects of the updating system on classification performance and updating dynamics, an offline analysis on features and dictionary size was accomplished using data acquired from two able-bodied subjects. While there were some instances of changes to classification performance, the primary difference was in the updating dynamics. By having an overdefined dictionary, less updates were required by the control and experimental classifiers. An increase in the dictionary size similarly suggests less updates being required as more historical data is preserved allowing the classifier to be more robust to limb-position. In all explored variants, the classifier with the updating system outperforms the control. This classification improvement is observed in the online experiment completed by three able-bodied subjects. While performance improvement is observed, it should be noted that the efficacy of the update is dependent on a subject's experience with myoelectric control and their ability to perform classes well. A system that incorporates the updating method with an large overdefined dictionary that has a controller delay within real-time constrants will illustrate classification performance improvement by spending only 3 seconds each time to update a troublesome class.

In conclusion, EASRC optimized with an information centric approach to updating shows promise in improving a subject's ability to perform hand grips. While there is a dependency on a subject's experience in updating dynamics, training a subject to utilize different musculature to provide more distinction in feature space may reduce the amount of time spent requesting updates. With the promising results from able-bodied subjects, we intend to increase the applicant pool to include amputees and hypothesize similar robustness to limb-position variations are likely to be observed in a clinical setting. We therefore expect our updating system to give greater prosthesis control to a user and ultimately improve their ability to perform activities of daily living.

References

Fougner, Anders, Oyvind Stavdahl, Peter J. Kyberd, Yves G. Losier, and Philip A. Parker (2012). "Control of upper limb prostheses: Terminology and proportional myoelectric controla review". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 20.5, pp. 663–677. ISSN: 15344320. DOI: 10.1109/TNSRE.2012.2196711.

Young, Aaron J., Levi J. Hargrove, and Todd A. Kuiken (2011). "The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift". In: *IEEE Transactions on Biomedical Engineering* 58.9, pp. 2537–2544. ISSN: 00189294. DOI: 10.1109/TBME.2011.2159216. arXiv: NIHMS150003.

Scheme, E and K Englehart (2011). "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use". In: *Journal of Rehabilitation Research and Development* 48.6, pp. 643–659. ISSN: 0748-7711. DOI: Doi10.1682/Jrrd.2010.09.0177.

Kuiken, Todd A., Guanglin Li, Blair A. Lock, Robert D. Lipschutz, Laura A. Miller, Kathy A. Stubblefield, and Kevin B. Englehart (2009). "Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms". In: *JAMA - Journal of the American Medical Association* 301.6, pp. 619–628. ISSN: 00987484. DOI: 10.1001/jama.2009.116.

Hargrove, L. J., K. Englehart, and B. Hudgins (2007). "A Comparison of Surface and Intramuscular Myoelectric Signal Classification". In: *IEEE Transactions on Biomedical Engineering* 54.5, pp. 847–853. ISSN: 0018-9294. DOI: 10.1109/TBME.2006.889192.

Farina, Dario, Aleš Holobar, Roberto Merletti, and Roger M. Enoka (2010). "Decoding the neural drive to muscles from the surface electromyogram". In: *Clinical Neurophysiology* 121.10, pp. 1616–1623. ISSN: 13882457. DOI: 10.1016/j.clinph.2009.10.040. URL: http://dx.doi.org/10.1016/j.clinph.2009.10.040.

Farina, Dario, Ning Jiang, Hubertus Rehbaum, Aleš Holobar, Bernhard Graimann, Hans Dietl, and Oskar C. Aszmann (2014). "The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges".

In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22.4, pp. 797–809. ISSN: 15344320. DOI: 10.1109/TNSRE.2014.2305111.

Smith, Lauren H., Levi J. Hargrove, Blair A. Lock, and Todd A. Kuiken (2011). "Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay". English (US). In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19.2, pp. 186–192. ISSN: 1534-4320. DOI: 10.1109/TNSRE.2010.2100828.

De Luca, Carlo J., L. Donald Gilmore, Mikhail Kuznetsov, and Serge H. Roy (2010). "Filtering the surface EMG signal: Movement artifact and baseline noise contamination". In: *Journal of Biomechanics* 43.8, pp. 1573–1579. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2010.01.027. URL: http://dx.doi.org/10.1016/j.jbiomech.2010.01.027.

Phinyomark, A, S Hirunviriya, C Limsakul, and P Phukpattaranont (2010). "Evaluation of EMG feature extraction for hand movement recognition based on Euclidean distance and standard deviation". In: *International Conference on Electrical Engineering/Electronics Computer Telecommunications and Information Technology 2010*, pp. 856–860.

Phinyomark, Angkoon, Pornchai Phukpattaranont, and Chusak Limsakul (2012). "Feature reduction and selection for EMG signal classification". In: *Expert Systems with Applications* 39.8, pp. 7420–7431. ISSN: 09574174. DOI: 10.1016/j.eswa.2012.01.102. URL: http://dx.doi.org/10.1016/j.eswa.2012.01.102.

Ortiz-Catalan, Max, Faezeh Rouhani, Rickard Branemark, and Bo Hakansson (2015). "Offline accuracy: A potentially misleading metric in myoelectric pattern recognition for prosthetic control". In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* 2015-November, pp. 1140–1143. ISSN: 1557170X. DOI: 10.1109/EMBC.2015.7318567.

Jiang, N, S Dosen, K R Muller, and D Farina (2012). "Myoelectric Control of Artificial Limbs: Is There a Need to Change Focus? [In the Spotlight]". In: *IEEE Signal Processing Magazine* 29.5, pp. 150–152. ISSN: 1053-5888. DOI: 10.1109/msp.2012.2203480.

He, Jiayuan, Dingguo Zhang, Ning Jiang, Xinjun Sheng, Dario Farina, and Xiangyang Zhu (2015). "User adaptation in long-term, open-loop myoelectric training: Implications for EMG pattern recognition in prosthesis control". In: *Journal of Neural Engineering* 12.4. ISSN: 17412552. DOI: 10.1088/1741-2560/12/4/046005.

Cornish, B. H., B. J. Thomas, and L. C. Ward (1998). "Effect of temperature and sweating on bioimpedance measurements". In: *Applied Radiation and Isotopes* 49.5-6, pp. 475–476. ISSN: 09698043. DOI: 10.1016/S0969-8043(97)00057-2.

Fougner, Anders, Erik Scheme, Adrian D.C. Chan, Kevin Englehart, and Øyvind Stavdahl (2011). "Resolving the limb position effect in myoelectric pattern recognition". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 19.6, pp. 644–651. ISSN: 15344320. DOI: 10.1109/TNSRE.2011.2163529.

Cipriani, Christian, Rossella Sassu, Marco Controzzi, and Maria Chiara Carrozza (2011). "Influence of the weight actions of the hand prosthesis on the performance of pattern recognition based myoelectric control: Preliminary study". In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 1620–1623. ISSN: 1557170X. DOI: 10.1109/IEMBS.2011.6090468. arXiv: NIHMS150003.

Biddiss, Elaine and Tom Chau (2007b). "Upper-limb prosthetics: Critical factors in device abandonment". In: *American Journal of Physical Medicine and Rehabilitation* 86.12, pp. 977–987. ISSN: 08949115. DOI: 10.1097/PHM.0b013e3181587f6c.

Biddiss, Elaine and Tom Chau (2007a). "Upper limb prosthesis use and abandonment: A survey of the last 25 years". In: *Prosthetics and Orthotics International* 31.3, pp. 236–257. ISSN: 03093646. DOI: 10.1080/03093640600994581.

Cordella, Francesca, Anna Lisa Ciancio, Rinaldo Sacchetti, Angelo Davalli, Andrea Giovanni Cutti, Eugenio Guglielmelli, and Loredana Zollo (2016). "Literature review on needs of upper limb prosthesis users". In: *Frontiers in Neuroscience* 10.MAY, pp. 1–14. ISSN: 1662453X. DOI: 10.3389/fnins.2016.00209.

He, Jiayuan, Dingguo Zhang, Xinjun Sheng, and Xiangyang Zhu (2013). "Effects of long-term myoelectric signals on pattern recognition". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8102 LNAI.PART 1, pp. 396–404. ISSN: 03029743. DOI: 10.1007/978-3-642-40852-6-40.

Sensinger, Jonathon W., Blair A. Lock, and Todd A. Kuiken (2009). "Adaptive pattern recognition of myoelectric signals: Exploration of conceptual framework and practical algorithms". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 17.3, pp. 270–278. ISSN: 15344320. DOI: 10.1109/TNSRE.2009.2023282. arXiv: NIHMS150003.

Simon, Ann M., Levi J. Hargrove, Blair A. Lock, and Todd A. Kuiken (2011). "A decision-based velocity ramp for minimizing the effect of misclassifications during real-time pattern recognition control". In: *IEEE Transactions on Biomedical Engineering* 58.8, pp. 2360–2368. ISSN: 15582531. DOI: 10.1109/TBME.2011.2155063. arXiv: NIHMS150003.

Scheme, Erik J., Bernard S. Hudgins, and Kevin B. Englehart (2013). "Confidence-based rejection for improved pattern recognition myoelectric control". In: *IEEE Transactions* *on Biomedical Engineering* 60.6, pp. 1563–1570. ISSN: 00189294. DOI: 10.1109/TBME.2013.2238939.

Chen, Xinpu, Dingguo Zhang, and Xiangyang Zhu (2013). "Application of a self-enhancing classification method to electromyography pattern recognition for multifunctional prosthesis control". In: *Journal of NeuroEngineering and Rehabilitation* 10.1, pp. 1–13. ISSN: 17430003. DOI: 10.1186/1743-0003-10-44.

Ams, Sebastian, Senior Student Member, Peter M Goebel, Ning Jiang, Bernhard Graimann, Liliana Paredes, Dario Farina, and Sebastian Amsuss (2014). "Self-Correcting Pattern Recognition System of Surface EMG Signals for Upper Limb Prosthesis Control." In: *IEEE Transactions on Biomedical Engineering* 61.4, pp. 1167–76. ISSN: 1558-2531. DOI: 10.1109/TBME.2013.2296274. URL: http://www.ncbi.nlm.nih.gov/pubmed/24658241.

Vidovic, Marina M.C., Han Jeong Hwang, Sebastian Amsuss, Janne M. Hahne, Dario Farina, and Klaus Robert Muller (2016). "Improving the robustness of myoelectric pattern recognition for upper limb prostheses by covariate shift adaptation". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24.9, pp. 961–970. ISSN: 15344320. DOI: 10.1109/TNSRE.2015.2492619.

Zhu, Xiangyang, Jianwei Liu, Dingguo Zhang, Xinjun Sheng, and Ning Jiang (2017). "Cascaded Adaptation Framework for Fast Calibration of Myoelectric Control". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.3, pp. 254–264. ISSN: 15344320. DOI: 10.1109/TNSRE.2016.2562180.

Chen, Xinpu, Xiangyang Zhu, and Dingguo Zhang (2009). "Use of the discriminant Fourier-derived cepstrum with feature-level post-processing for surface electromyographic signal classification". In: *Physiological measurement* 30.12, p. 1399.

Betthauser, Joseph L., Christopher L. Hunt, Luke E. Osborn, Matthew R. Masters, Gyorgy Levay, Rahul R. Kaliki, and Nitish V. Thakor (2018). "Limb Position Tolerant Pattern Recognition for Myoelectric Prosthesis Control with Adaptive Sparse Representations from Extreme Learning". In: *IEEE Transactions on Biomedical Engineering* 65.4, pp. 770–778. ISSN: 15582531. DOI: 10.1109/TBME.2017.2719400.

Myers, Raymond H and Raymond H Myers (1990). *Classical and modern regression with applications*. Vol. 2. Duxbury press Belmont, CA.

Cao, Jiuwen, Kai Zhang, Minxia Luo, Chun Yin, and Xiaoping Lai (2016). "Extreme learning machine and adaptive sparse representation for image classification". In: *Neural Networks* 81, pp. 91–102. ISSN: 18792782. DOI: 10.1016/j.neunet.2016.06.001. URL: http://dx.doi.org/10.1016/j.neunet.2016.06.001.

Vita

Teja Karri was born on June 5th, 1996 in Hyderabad, India. He completed his bachelors of science in biomedical engineering from Boston University in May 2017. He is presently part of Tau Beta Pi (Engineering Honor Society) and Alpha Eta Mu Beta (Biomedical Engineering Honor Society). Teja will be completing his Masters in Science and Engineering degree in Biomedical Engineering from Johns Hopkins University in May 2019. In his undergraduate research Teja worked in a synthetic biology on a high throughput turbidostat system for assessing bacterial evolution. Interested in a different problem space, Teja came to Johns Hopkins to work with Dr. Nitish V. Thakor on Brain-Computer Interfaces and upper-limb prosthesis. Here he focused on an updating system to improve motor control.